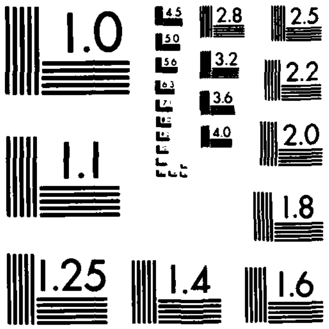
AD-A172 919   ROBOTIC MANIPULATOR CONTROL PERFORMANCE EVALUATION(U)   1/3
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH
M B LEAHY AUG 86 AFIT/CI/NR-86-173D

UNCLASSIFIED   F/G 13/9   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 86- 173D	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Robotic Manipulator Control Performance Evaluation		5. TYPE OF REPORT & PERIOD COVERED ~~THESIS~~/DISSERTATION
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR*(s)* Michael B. Leahy, Jr.		8. CONTRACT OR GRANT NUMBER*(s)*
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: Rensselaer Polytechnic Institute		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 1986
		13. NUMBER OF PAGES 206
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASS
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-1

Lynn Wolaver 23 Sep 86
LYNN E. WOLAVER
Dean for Research and
Professional Development
AFIT/NR

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

ATTACHED ...

86 10 10 107

173

ROBOTIC MANIPULATOR CONTROL PERFORMANCE EVALUATION

by

Michael B. Leahy Jr.

A Thesis Submitted to the Graduate
Faculty of Rensselaer Polytechnic Institute
in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY
Major Subject: Electrical Engineering

Approved by the
Examining Committee:

George N. Saridis, Thesis Advisor

Stephen J. Derby, Member

Alan A. Desrochers
Alan A. Desrochers, Member

Robert B. Kelley
Robert B. Kelley, Member

Rensselaer Polytechnic Institute
Troy, New York

August 1986
(For Graduation December 1986)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

DEDICATION

This disseration is dedicated, with love, to my wife Mary Jean, my parents Michael and Joan, and my young son Michael.

ACKNOWLEDGEMENTS

I would like to extend my sincere gratitude to Dr. George Saridis for his advice, ideas and support. I would also like to thank the members of my doctoral committee, Dr. Alan Desrochers, Dr. Robert Kelley, and Dr. Steven Derby, for their guidance.

To everyone in the RAL who lent a hand when things looked bleak, I extend a harty thank-you. I would especially like to thank four friends; Chris Seaman and Bob Balaram who welcomed me to the lab and provided much needed guidance and support during my first year, Kimon Valavanis for his advice and inspiration and, Steve Murphy who was always there when I needed someone to talk to and whose editorial skills I exercised fully in the preparation of this document.

I would also like to thank Alessandro De Luca for the lengthy discussions which broadened my horizons both intellectually and culturally.

A special thanks to Chris and Steve for their assistance in renovating my first house, don't make the same mistakes twice.

Finally I would like to thank Trisha and Teal for their equation typing, figure labeling, and all their help in putting together the RAL reports and conference papers that proceeded this paper.

ABSTRACT

A robotic manipulator dynamically based controller performance baseline is established by the creation and utilization of a hierarchical robotic evaluation environment. Creation of a hierarchical robotic evaluation environment provides an original solution to the problems that previously constrained real-time evaluation of modern manipulator control schemes. Utilization of that environment fulfills the critical robotic research requirement for experimental application of proposed theories. The performance baseline is established by simulated and experimental evaluation of feedforward dynamics and feedback loop design for joint motion high speed trajectory tracking robot control. The real-time performance produced by application of all proposed robotic control techniques to harmonic and gear driven manipulators can be extrapolated from the baseline. A feedforward loop composed of uncoupled inertia and gravity dynamics exhibited the best tracking accuracy. Forces unmodeled by those dynamics can be effectively treated as disturbances to the feedback loop. Dynamic based control techniques exhibited the potential to control high speed gross motion of a manipulator without additional sensor devices. A theoretical basis for calculation of calibration uncertainties has been developed to assist the further evaluation and integration of modern control techniques into a hierarchical intelligent control system.

CHAPTER 1

INTRODUCTION

1.1 Motivation

Industrial manipulators are currently controlled by individual single joint PD and PID feedback loops [61]. Those methods are adequate for slow repetitive motions that can be programmed off-line or taught to the manipulator. However, they are inadequate for implementation at the hardware level of a hierarchically intelligent machine ([105],[86]) operating autonomously in an uncertain environment. The motivation for this research is the search for a gross motion robotic manipulator control scheme whose performance is suitable for implementation in such a hierarchically controlled intelligent machine.

1.2 Objective

Performance evaluation of currently proposed manipulator control techniques would reduce the search for a gross motion control scheme applicable to intelligent machines. The objective of this research is the establishment of a dynamically based controller performance baseline by the creation and utilization of a hierarchical robotic evaluation environment.

1.3 Problem Statement

A major deficiency in current robotic manipulator control research has been the lack of thorough real-time evaluation of the proposed algorithms. Most results published in the literature are simulation studies of algorithm performance over one arbitrary trajectory [56]. Knowledge of currently proposed control techniques effectiveness in real-time environments would be invaluable for modifications of existing control strategies and design of new control methods.

Real-time evaluations have been principally constrained by the following factors:

1. the speed of the computers originally supplied with existing manipulators is insufficient for the degree of computation required,
2. the programming languages developed for robot arms lack the necessary flexibility,
3. the manipulators may have existing hardware control loops that must be bypassed or accounted for.

Development of a system which eliminates these problems and thus permits real-time testing is necessary to advance the state of the art in robot control methodology.

In conjunction with the development of modern robotic control methods have been the efforts of Saridis [86-7], Valavanis [105] and others to develop mathematical theories for the architecture and control of a hierarchical robotic work cell control. Those mathematical theories will govern selection and application of manipulator control techniques. Further development and testing of those theories requires real-time robotic control systems organized in a hierarchical fashion.

The intelligent work cell hierarchy must select appropriate manipulator control methods based on operational task and environment. A missing link in that development is the ability to convey information about the level of precision in manipulator end-effector position and orientation to the upper levels of the hierarchy. Knowledge of manipulator end-effector position and orientation is clouded by uncertainty. A major source of that uncertainty is manifested in manipulator calibration. Calculation of that uncertainty is required so that the hierarchy can compensate for the lack of precision by selecting an appropriate controller.

Knowledge of calibration uncertainty also enables the segregation of manipulator uncertainties. The effects of inertial parameter and load uncertainty on controller performance must be separated from calibration induced effects. Knowledge of the effects of operational environment aberrations such as parameter and load variations on controller performance are essential for an intelligent decision process.

1.4 Method Of Approach

The problems stated above are addressed in a sequential fashion. Each new development forms the foundation for the next step in the evaluation of gross motion robotic manipulator control for implementation in a hierarchical intelligent work cell. The first step is the development of a suitable evaluation environment.

An original solution that eliminates the real-time evaluation constraints detailed in the previous section is presented by creating a hierarchical robotic evaluation environment. That solution requires the development and integration of three major components: a hierarchical manipulator control system, customized efficient algorithms for computation of manipulator dynamics, and software libraries that support simulation and real-time modern control algorithm performance evaluation. The principle of decreasing intelligence with increasing precision is applied to the design and implementation of a three stage hierarchical control system for a six degree of freedom PUMA manipulator. The PUMA dynamical formulations are studied and symbolically reduced to produce the necessary efficient dynamics algorithms. Modular software is developed in PDP assembly language and VAX FORTRAN to empower the hierarchy with the ability to simulate, control, and analyze the effectiveness of proposed manipulator controllers. The developed environment is utilized to conduct two case studies that reduce the scope of the search for control techniques suitable for the intelligent work cell of the future.

Manipulator dynamics play a vital role in arm simulations and numerous proposed modern robotic control techniques ([56],[77],[84]). Therefore the logical first step in real-time control implementation research is an evaluation of the effects of dynamics on robot control. The computed-torque technique ([77-8],[57]) provides a mathematically well defined, dynamically dependent basic control algorithm for the study of the effects of dynamics on real-time robotic control. The performance of the computed-torque algorithm using four forms of dynamics in the feedforward loop is evaluated by both computer simulation, and real-time implementation over six different operational configurations. The dynamical formulations employed in the controller are: complete Newton-Euler, and three reduced forms of Lagrange-Euler: full, block, and diagonal inertia matrix, all with gravity but without Coriolis and centrifugal terms. By evaluating PUMA manipulator performance variations the effects of computed-torque feedforward loop neglected dynamics on gross motion joint control are exposed.

In the first case study computed-torque control technique efficacy is proven insufficient for gross motion control of the PUMA. Forces unmodeled by Lagrange-Euler techniques are a vital component in the actual dynamics of a PUMA manipulator. Their effects, especially on the small links, are too pervasive to be eradicated by the computed torque feedback loop employed in chapter four. Therefore, the second case study investigates modifications to the original computed-torque law that provide compensation of unmodeled manipulator dynamics. New inertial parameters [94] that better model the actual arm dynamics are evaluated. Friction compensation is implemented by an additive torque switching function whose limits have been defined by a performance characterization of the PUMA [43]. The bandwidth of the PD feedback loop is increased. A PID feedback loop is inserted in place of the PD loop. The results from evaluation of those modifications, over operational configurations identical to the first case study, are compared and contrasted to gain insight about the optimum method for compensation of unmodeled forces. Those evaluations reveal the capabilities of non-sensor based controllers to compensate for unmodeled PUMA forces.

An original solution to the calibration uncertainty calculation problem is developed. A theoretical basis for the employment of the Entropy function as a measure of calibration uncertainty is established. The selection of the Entropy function incorporates uncertainty information into the mathematical formulation of an intelligent work cell ([105],[86]) and permits the evaluation of the ramifications of environmental aberrations on gross motion manipulator control methods.

1.5 Contribution And Summary Of Results

The main contributions of this research are as follows:

1. An original solution to the constraints restricting real-time implementation of modern control techniques on a six degree of freedom revolute manipulator.
2. Evaluation of dynamic models for simulated and real-time control of a six degree of freedom revolute geared manipulator.
3. Evaluation of feedforward and feedback techniques for compensation of six degree of freedom revolute geared manipulator forces unmodeled by Lagrange-Euler dynamical models.
4. Development of the theoretical basis for application of the Entropy function as a measure of manipulator calibration uncertainty.

These efforts have significantly enhanced the real-time manipulator control database while providing the control system foundation essential for continued development of an intelligent machine.

1.6 Organization

This dissertation consists of seven chapters organized as follows: Chapter two reviews the literature pertinent to the creation and application of the hierarchical robotic evaluation environment. Chapter three documents the developmental history of that environment and presents an overview of the main components. Chapter four presents the research and conclusions on evaluation of dynamic models for robot control. Chapter five evaluates the effects of feedforward and feedback compensation techniques on efficacy of a PUMA under computed-torque control utilizing the optimum dynamic model of chapter four. The theoretical basis for the utilization of the Entropy function as a measure of manipulator calibration uncertainty is developed in chapter six. Conclusions and suggestions for future research are presented in Chapter seven. Appendices contain evaluation data not presented in the chapters.

CHAPTER 2

LITERATURE REVIEW

In this chapter research pertinent to the development and application of a hierarchical robotic evaluation environment is reviewed. Research published prior to the development of that environment is grouped into sections on; real-time control systems, performance characterization, and efficient dynamics. Earlier efforts on real-time joint space gross motion control are reviewed. Previous research on techniques to quantify uncertainty in manipulator calibration is the final area of review.

2.1 Real-time Control Systems

The first study of the developmental issues of a real-time control system at the Rensselaer Polytechnic Institute Robotics and Automation Laboratory (RAL) was completed by Valavanis in May of 1983 [102] and advocated breaking away from VAL. The manipulator system studied at that time was the PUMA-600 installed in the lab. The PUMA-600 is a six degree of freedom revolute manipulator mechanically similar to the PUMA-560. A PUMA-600, like most industrial robots [61] is equipped with its own dedicated controller and programming language. A thorough description of the original control system was presented by Valavanis [102]. The PUMA-600 control system is composed of an LSI-11/02 which interfaces to the six joints via six Motorola

6503 microprocessors [100]. Control of the robot is accomplished through the VAL programming language [101-2]. That system does not allow interfacing with other robots and has the following disadvantages:

1. the inability to close the control loop back to the computer level,
2. the limited computational power of the LSI-11/02, and
3. the fixed 28ms sampling intervals.

Those limitations prohibit employment of the existing PUMA control scheme for testing of modern control methodology.

The PUMA-600 real-time control system first proposed by Valavanis consisted of the following hardware modifications:

1. replace the LSI-11/02 with an LSI-11/23,
2. connect the Q-bus of the LSI-11/23 to the unibus of a VAX-11/750,
3. increase the PUMA system memory size to 128KB,
4. modify the hardware to permit velocity and acceleration feedback to the control computer.

Valavanis proposed the creation of a robotic language based on the code in the joint microprocessors, which would include communication protocol, so that the user could break away from VAL. The driving force behind this proposal was a desire to duplicate the suboptimal manipulator work done by Saridis and Lee [88], on a MIT arm at Purdue, on the RAL

PUMA.

The proposed system was an ambitious plan beyond the scope of the original researchers immediate goals. The decision was made to design and implement an intermediate level system that would further enhance the knowledge framework required to develop the proposed hierarchical system. The first working RAL real-time control system was developed by Valavanis, Walter and Leahy ([49],[50],[104]) and was operational in 1984. Their approach created a dedicated non-hierarchical control system by disconnecting the existing computer controller and connecting a VAX-11/750, running under the VMS operating system, in it's place. The VAX communicated with the existing hardware level by a DR11-W DMA link to the arm interface board. That system broke away from VAL and duplicated the control commands with a library of modular FORTRAN subroutines. The trajectory planning and inverse kinematic functions of VAL were not required and therefore not duplicated. The VAX/PUMA system had several advantages over the existing controller:

1. the superiority of the VAX-11/750 over the LSI 11/02 used by VAL,
2. the control loops are closed back to the computer level,
3. the joint microprocessors are employed only as buffers bypassing the supplied control loop, and

4. the increased power and flexibility of FORTRAN over VAL.

The VAX/PUMA control system enabled the first real-time evaluations of several modern joint space robotic control techniques [104].

Researchers at Unimation lead by Shimano sought to overcome the problems of the VAL language by the development of VAL-II [91]. VAL-II was designed to support network communications, real-time trajectory modification, general sensory interfaces and concurrent user-program execution. PUMA-560 manipulators are equipped with VAL-II. The original PUMA hardware is modified by installation of a LSI-11/23 as the host computer and a connection to a supervisory system through a local network via RS-232C serial lines. The addition of the LSI-11/23 allows for complex algorithm programming due to its floating point capabilities. The new language also provides standard arithmetic operations, operators, and control structures commonly found in high-level "structured" computer languages. The path that the manipulator is instructed to follow can be altered in real-time but only by the addition of cartesian data at 28ms intervals. Even with all its improvements VAL-II still is unsuited for testing of modern control techniques.

The topic of robot language has been very active. Development of other languages for real-time control systems proceeded in parallel with VAL-II. Gruver, Soroka et.al. review the capabilities of commercially available languages

[25]. Their developers were primarily concerned with easing the program task for robotic system users and not with evaluation of modern control techniques. Most languages assume that the controller is a fixed element using standard industrial PD control techniques. Like VAL-II those languages support the industrial environment but not the research environment for modern controls.

Researchers at Purdue, University of Toronto, and JPL are also active in real-time control research employing PUMA arms. They created their own real-time control systems and languages at the same time that the RAL VAX/PUMA system was under development. The JPL system [7] is part of larger hierarchically based control station which includes TV monitors, a graphics system for informational display, alarm sound generators and control mode switches. Like the RPI system the JPL PUMA controller connects directly to the arm interface board, bypassing the LSI-11/02 and VAL. Unlike the RAL system their PUMA-560 is controlled by a NS 16000 microcomputer.

Under the direction of Goldenberg researchers at the University of Toronto have designed a non-hierarchical PUMA control system [59]. That system is characterized by the original PUMA controller LSI-11/02 running under an RT-11 operating system with a serial connection to a PDP-11/23 Plus system. Programs are developed on the PDP and sent to the 11/02 for execution. There is no other provision for inter-system communication. Most system software is written in FORTRAN and the RT-11 kernal software had to be modified so that interrupts could be processed correctly. The software is based on modules not subroutines and is written inefficiently. The limitations of using an operating system in a real-time controller coupled with the slow speed and lack of full floating point support of the 11/02 greatly restrict the user's ability to implement modern control algorithms with this system. In fact only primitive single joint movement algorithms have been tested.

Meanwhile, at Purdue, Hayward, Paul and others [26] were developing RCCL: A Robot Control "C" Library. As was the case for the VAX/PUMA system, RCCL is not a language but a series of subroutine calls that allow control of a robotic manipulator. After that point the similarity ends. RCCL is written in the "C" language and runs under a UNIX operating system equipped with specialized real-time device drivers and kernel code modifications. RCCL completely duplicates and expands on the functions provided in most robot languages while allowing development of modern control techniques. Under RCCL control four processes are executing concurrently. The user process runs under the time shared environment on their VAX-11/780 and executes the user's "C" algorithm containing RCCL subroutine calls. A motion request queue allows the user process to communicate with the next level, called the setpoint process. The lowest level is the servo process which controls the position or torque of the manipulator. The key to the real-time capabilities of the system is the real-time communications channel which allows communication between the servo and setpoint process at speeds suitable for real-time modern control. Both a Stanford arm and a PUMA-600 are currently controlled by RCCL. In a break from the design philosophy of the RAL and JPL systems the PUMA servo process still employs the LSI-11 to supervise the joint microprocessors and perform the arm calibration. Although potentially powerful enough to support testing of all proposed real-time control techniques the RCCL system has been designed to support cartesian control algorithms that employ force and torque control. Published results that utilized this system have only involved forms of cartesian force control ([5],[58]).

The National Bureau of Standards has also developed a real-time control system (RCS) ([6],[67]) composed of four levels. RCS serves as a major subsystem in their Automated Manufacturing Research Facility. Work at NBS is driven by a desire to create guidelines for standardizing interfaces to robots for easy implementation of a hierarchical control system. As in VAL-II, communication between the the arm controller and the external computer is by an RS232 communication link. The four levels of the RCS hierarchy subdivide a general instruction in the manner of increasing precision with decreasing intelligence as proposed by Saridis

[86]. RCS supports the types of motion control commonly seen in industrial robots. The manipulator employed as a test case is a Cincinnati Milacron T3 employing the existing servo control but modified to allow control by the RCS. The inability to bypass the existing servo control, coupled with the lack of computational complexity and flexibility prohibits the performance of modern robotic control algorithm evaluation on the RCS.

In the same time frame as the other system developments, researchers at Georgia Tech have been creating a hierarchical control system for the study of the methodology for the coordinated control of two robot arms [1-2]. Their system is built around 16 Intel 8086 microprocessors and an Intel 8089 I/O processor. The two robot test case is composed of a PUMA-550 and a PUMA-560. Their design takes the approach followed by JPL and RPI one step further by replacing even the joint microprocessors. Each joint now has an axis computer defined as an 8086-based system in minimum mode. Each arm has two prediction computers composed of 8086 based systems in maximum mode. The prediction computers plan the trajectories and send joint commands to the axis computers. The top level of the hierarchy is called the Multi-Arm Coordination Computer, MACC, which is an 86/14 single card computer including RAM, a numeric processor and a fixed/floating point arithmetic processing unit. The MACC executes a coordination algorithm to calculate the new desired slave arm position based on information from the prediction computers. Trajectory modifications are sent to the five slave axis computers by the I/O processor. The system is still in the developmental stage and only control via conventional servo techniques has been applied. Even for that simple program the transfer of command modifications to the appropriate axis computers required 8ms with total sampling times of 20ms. Implementation of modern single arm control techniques was considered beyond the scope of the current research efforts but the system seems to have the potential to support such work.

The amount of work in the area of real-time control systems has expanded greatly within the last year. Researchers at Ohio State University are developing operating system primitives and a real-time control system for a six legged robot ([76],[89]). As in the case of the Georgia Tech system, 16 8086 based single board computers are being hierarchically arranged. The operating system under development is called GEM: generalized executive for real-time multiprocessor applications. Employment of this system for study of real-time modern control was suggested but has not been implemented.

Two groups of researchers are examining the design of manipulator control systems based on arm dynamics. Niagam and Lee [73] researched the topic of cost effective architectures using currently available microprocessors to compute real-time joint torques based on dynamics. They performed a feasibility study to verify that special purpose architectures are required to meet their sampling rate constraint of 3ms in a cost effective manner. Their proposed architecture uses six microprocessors to accomplish the efficient computation of the Newton-Euler dynamics with a seventh processor used to coordinate host communications. That proposal is a refinement of the earlier work of Lee, Mudge and Turney [57] who proposed development of a special purpose processor for accomplishing the same task. In both cases the controller functions as an attached processor controller in a hierarchical environment. The proposed systems were not operational at the time of their last report.

Zheng and Chen designed a loosely coupled multiprocessor system based on dynamical control of a manipulator [112]. The multiprocessor system is composed of a PDP-11/23 serving as the central processor and PDP-11/03's employed as satellite processing units. The computers communicate with each other over DEC DRV11 parallel interface modules. Satellite processors are able to communicate with each other directly. Their work on the Newton-Euler state space equations produced a computation scheme which allows the individual joint forces and moments to be calculated in parallel on satellite processors and then sent to the central processor for computing the final torques. A two link robot was used to test a FORTRAN program that computed the applied torques. Use of the central and two satellite processors did decrease the computational time by half. However, the computational time of 47.1 ms is still excessive and is a function of the limited speed of floating point calculations especially in the LSI-11/03's which only have the DEC floating point instruction set [81], not full floating point capability. That basic limitation will prevent even assembly language code from executing efficently.

Turner is leading another group of researchers interested in multiprocessor real-time control systems [99]. They have simulated a four microprocessor system designed to allow either force or position control at high servo rate speeds. In a break from most other controller designers they propose that single microprocessors not be employed to control each joint. They claim that single joint microprocessors impose severe constraints for multi-input multi-output control strategies, an example being the 5.8ms of interprocessor communication time in the system designed by Zheng.

Turner's system consists of language, dynamics and servo processsors and a path planner. The PDP-11/60 language processor handles executive, file manager, interpreter and other high level functions along with interfacing to the outside world. The unique part of the system is the dynamics processor which calculates the joint couplings for control purposes in parallel with the path planner. Both the path planner and dynamics processor are 32 bit bit/slice processors. The servo interface is a 16 bit bit/slice machine. Like the RCCL system only modern force control algorithms can be tested on this system. The position control mode is too crude for implementation of modern position control techniques.

Turner's simulation results are quite impressive but their control scheme is nothing more than a sophisticated VAL scheme. In place of the hardware PD loop they implement Salisbury's active stiffness controller [82]. The parameters used by that controller are all calculated in open-loop fashion from the desired values. Inertial, gravitational and coordinate transformation values don't reflect actual arm position. By calculating the open-loop values in parallel with the servo loop the authors are able to produce their high speed values, but as with VAL the loop is not closed to the computer level.

Dupourque makes a case for controlling six degree of freedom robots with no external hardware and a single processor [18]. His "Robot Operating System" uses a 16 bit Motorola 68000 running at 8mhz and an interface card to receive and send analog inputs and outputs. That system has no floating point functions so extensive use is made of look-up tables. In its current state of development this system can not support real-time evaluation of modern robotic controllers.

2.2 Performance Characterization

In order to conduct worst case real-time tests, worst case data must be available. In their first attempt at real-time control of the RAL PUMA-600 the authors of [104] arbitrarily selected a set of trajectories to track, that forced each joint to traverse 90 degrees in 1.5 seconds. No data existed to validate whether in fact the peak speeds required by those trajectories where reachable by every joint. If the speeds were unobtainable then velocity tracking would be impossible. Analysis of the velocity error results from that study reveal that several joints peak velocities were well below the desired. Another important aspect of PUMA-600 performance that needs to be categorized is the effects of friction. Simulation studies of algorithm performance commonly ignore the effects of friction in their analysis. Friction is a powerful force on a PUMA. A means of accounting for friction may be necessary to obtain satisfactory real-time performance of certain control methods ([55],[64],[78],[107],[109]).

In order to answer these questions a performance characterization of the PUMA-600 was undertaken in the RAL [43]. Step input tests were performed to find the minimum current levels required to just overcome the effects of friction. Gear friction was found to vary along the gear train and not to contain distinct static and dynamic components. The proposed friction compensation model was a switching function which adds a constant value to the uncompensated current in that direction. The variations in friction value must be modeled as uncertainties in the operational environment.

In real-time control articles presented at the 1985 IEEE International Conference on Robotics and Automation the issue of friction compensation of a PUMA was addressed and Backes, Leininger and Chung [5], and Zhang and Paul [111] model friction in an identical fashion. Also at that conference Mukerjee and Ballard presented a tabular approach to the friction problem [70]. To support open loop control they proposed modeling manipulator friction as the sum of coulomb, viscous, and transmission components and tabulating these values in separate look-up tables searched by measured forces, moments and positions. That approach requires the installation of additional measurement devices to produce the advantages of ordinary closed loop control.

The maximum achievable individual joint velocities were found by applying the maximum current to each joint motor and measuring the steady state velocity. Peak velocities from the first real-time study were found to be unrealizable for the large joints. Small joint velocities can exceed 4.0 rads/sec. Armed with this information a set of realistic maximum joint trajectories can be derived.

2.3 Efficient Dynamics

All non-adaptive modern robotic control techniques utilize some form of manipulator dynamics in their control laws. A major stumbling block in the drive for real-time implementation has been the computational complexity of these formulations [28]. Bejczy and Paul were the first to employ symbolic state equation techniques to reduce the complexity of the Lagrange-Euler formulation [9]. By geometric and numeric evaluation of the symbolic Lagrange-Euler equations they significantly reduced the computational requirement for a Stanford-Scheinman arm with fixed wrist, and provided insights into general reduction techniques. Bejczy and Lee expanded upon the brief presentation in [9] by discussing ways of reducing the complete Lagrangian formulation through matrix, vector, numerical and significancy analysis [8]. A reduced set of equations for a specific manipulator was not presented.

The most common technique suggested for reducing dynamical complexity is the elimination of the Coriolis and centrifugal terms in the Lagrangian formulation ([104],[55]) since they require the greatest computational burden [28]. A further simplification commonly suggested is to diagonalize the inertial matrix. Paul, Zong and Zhang derived a reduced set of equations for PUMA-600 diagonal inertial and gravity vector values using significance analysis [79]. Those equations were accurate to within ten percent for inertial values and one percent for gravity vector terms.

In a recent work Isaguirre and Paul proposed a set of equations for computation of the inertial and gravitational coefficients based on the relationships between the links [32]. A reduced model for the inertial matrix is also proposed. The reduction of the unloaded PUMA-600 equations to 65 multiplications and 41 additions is impressive. However, the effects of a load are included as additive terms to the unloaded equations, resulting in an additional 160 multiplications and 101 additions.

Analysis of the effects of their reduced equations on coefficient accuracy, and reduced model on interjoint coupling are not thoroughly discussed. Paul's model for the PUMA is different than the one used in the RAL so the principles behind his work are applicable but the equations are not exact.

Another reduction approach is to tabularize the values, store them in memory and employ various forms of table look up to determine the needed values. Hollerbach discusses the problems associated with these approaches and also proposes a recursive formulation for the Lagrangian formulation [28]. Use of full Lagrangian dynamics for real-time control is still not within the power of existing systems.

For our studies the need for complete dynamics could be satisfied by the Newton-Euler formulation first proposed by Luh, Walker ana Paul [63]. Although much more compact than the Lagrangian formulation the basic Newton-Euler computations still require 852 multiplies and 738 additions [28]. The number of computations could be simplified by accounting for the structure of the specific manipulator.

Hollerbach and Sahar discussed the reductions in Newton-Euler formulation computational complexity possible for manipulators with spherical wrists [29]. Their reduced Newton-Euler formulation for a six degree of freedom rotary manipulator with no offsets and a spherical wrist requires 408 multiplications and 324 additions. By also assuming precomputed inverse kinematics and simplified inertial parameters the computations were further reduced to 194 multiplications and 138 additions. Kanade et. al., expanded on that work in search of a set of custom Newton-Euler equations for their direct drive II arm [33]. The computational savings from a series of generic reduction measures is clearly presented. Newton-Euler equations for a general six degree of freedom rotary manipulator with parallel/perpendicular axis, spherical wrist and diagonal inertia require 393 multiplications and 305 additions. Additional reductions depend entirely on the specific structure of the manipulator.

Horak developed a general manipulator full dynamics symbolic model that combines the most desirable aspects of both Lagrange and Newton-Euler formulations in a fast computational scheme [30]. Lagrange-Euler non-matrix equations are utilized to model the first three links. The recursive Newton-Euler formulation is employed to model the wrist. Application of this method to a Stanford arm produced a full model with 361 multiplications and 256 additions. The authors's claim of an additional factor of two computational savings, over Luh's [63] method, from application of good coding techniques is invalid since the same principles were not applied to both algorithms.

Since the control system proposed in this thesis is only capable of sequential computations the recent efforts in parallel computation of dynamics ([62],[75]) will not be discussed.

2.4 Real-time Results

Although a large body of simulation knowledge has been created, studies of real-time performance have been scarce. Only researchers at:

1. Rensselaer Polytechnic Institute, RPI
2. Massachusetts Institute of Technology, MIT
3. Carnegie-Mellon University, CMU

4. University of California at Davis, UC Davis,
5. Tokyo Institute of Technology

have presented results from attempts at real-time joint space control of a robotic manipulator using modern techniques.

At the Tokyo Institute of Technology Furuta, Kosuge and Yamakita [21] have applied a nonlinear feedback technique which allows the design of optimal control law with quadratic constraints to a PT-300 manipulator. The PT-300 is mechanically equivalent to the GCA/DKP 300V manipulator. The optimal control formulation produced a PID control law that has been employed to control the three positioning degrees of freedom over one slow trajectory.

At UC Davis, Anex and Hubbard have applied an adaptive control technique to a RHINO XR-2 manipulator [4]. The adaptive technique studied was proposed by Horowitz and Tomizuka [31]. Several modifications to that algorithm had to be made to allow for real-time control testing. The results presented were from simple, slow, single joint motion of the bottom two links. Those specific findings are flawed by the author's claim of a 530 Hz control frequency while only calculating the velocity every 93.4ms. The observations about real-time implementation derived from those preliminary findings should help in future evaluations of adaptive control techniques.

The first real-time joint control results for a six degree of freedom manipulator were published by Valavanis, Leahy and Saridis [104] in 1985. That work was completed by the summer of 1984 and consisted of an evaluation of four modern control algorithm's effectiveness in tracking one trajectory with an unloaded PUMA-600. Simulation and real-time results were compared and contrasted. The VAX/PUMA system described earlier was employed for real-time testing. The four techniques tested were:

1. computed torque with simplified dynamics,
2. computed torque with complete dynamics,
3. the optimal/PID technique developed by Luo [66], with simplified dynamics, and
4. adaptive control using perturbation equations of motion developed by Chung and Lee [14].

The general conclusions reached by that study are valid but the specific results were flawed by the following:

1. the fastest sampling speed of 50ms is unacceptable for real-time control of a PUMA with any degree of accuracy,
2. the varying sampling speeds between the different algorithms made comparisons difficult if not invalid,
3. only one operational configuration was tested.

That work was designed as a case study of the VAX/PUMA control system, not an exhaustive evaluation. Further research at the RAL revealed that conceptual and coding errors existed in the Newton-Euler dynamics, the optimal/PID, and adaptive algorithms. Characterization studies of the arm also showed that the velocity trajectory was unachievable [43]. Although flawed, the first attempt at real-time joint space control does provide excellent insights into the problems associated with real-time implementation of modern control methods.

The 1986 IEEE International Conference on Robotics and Automation witnessed the publication of three new papers in the area of real-time joint space control. Leahy, Saridis and Valavanis presented a study on the effects of dynamics on robotic control [52]. The errors in their earlier research [104] were corrected by creation of a hierarchical robotic evaluation environment. That paper discussed the application of a computed-torque technique employing four forms of feedforward loop dynamics. Actuator inertias were not considered and small link torque to current conversion factors were altered based on experimental data to enable the small links to track the desired trajectory. The effects of neglected dynamics on simulated and real-time performance were clearly illustrated. The author's claimed that utilization of uncoupled dynamics in the feedforward loop produced the best overall control algorithm performance and that simulation results did not accurately predict real-time performance.

Although those simulation results are accurate the real-time data is invalidated by implementation errors. The dynamic model and control system reference frames were 180 degrees out of phase for links 1 and 3. Also the conversion factors utilized to convert torques to motor currents incorrectly modeled the gear ratios and A/D convertor current to counts ratio. Subsequent research has corrected both of those errors.

At MIT An, Atkeson, and Hollerbach investigated the application of feedforward control to the MIT direct drive arm [3]. The primary purpose of that study was to verify their inertial parameter estimation technique. They demonstrated that velocity feedback is an essential part of any manipulator control law and that the addition of feedforward complete manipulator dynamics had a significant impact on tracking accuracy. Feedforward dynamics improved the performance of the first two links. For the light third link unmodeled dynamics like friction became significant and reduced the role of the feedforward terms.

Researchers at CMU conducted an evaluation of computed torque performance on the CMU direct drive II manipulator [35]. Like the RPI study the effects of neglected dynamics on algorithm efficacy were evaluated. Like the PUMA, the direct drive II manipulator is sensitive to the dynamic model employed in the feedforward loop. However, the inclusion of more complete dynamics enhances direct drive arm performance. Incomplete manipulator modeling lead to large tracking errors in this study. Both inertial coupling and Coriolis and centrifugal forces have a significant effect on manipulator performance. That result is due to the lack of friction and gearing effects coupled with the subsequent higher velocities achievable by a direct drive manipulator. Excellent tracking performance was achieved by individually tuning the feedback gains for each joint and a 2ms sampling time. That research verified that a direct drive manipulator is an excellent approximation of the Lagrange-Euler dynamics model previously utilized in modern manipulator control algorithm simulation studies [56].

2.5 Calibration Uncertainty

In modern industrial applications the manipulator is an integrated component in a work cell consisting of fixtures, part transportation systems and other robots. All of those devices are designed and calibrated independently. Efficient work cell performance requires the determination of the exact position and orientation of the manipulator. The procedure to determine manipulator position and orientation is dominated by uncertainty.

Brooks defines the three major sources of uncertainty in a manipulator based work cell as [11]:

1. the manipulator,
2. the objects to be manipulated, and
3. the introduction of these objects into the work environment.

To quantify the ramifications of inertial parameter and object uncertainty on the repeatability of modern gross motion control algorithms knowledge of the manipulator uncertainty is essential.

Most manipulators require a calibration procedure to align the individual joints with some external reference frame common to all work cell components. Manipulator uncertainty is a function of the resolution of the joint positioning system instrumentation and errors produced by that calibration procedure. Calibration is generally based on an ideal set of kinematic parameters. Uncertainty in manipulator calibration is primarily the consequence of manufacturer's tolerances in robot fabrication. Those tolerances introduce uncertainty into the values of the kinematic parameters utilized in manipulator calibration. That uncertainty is reflected into the alignment of the individual joints with the common reference frame. End-effector position is dependent on the individual joint angle values. Consequently, calibration uncertainty results in uncertain knowledge of the absolute end-effector position.

Previous research on calibration uncertainty has centered on its elimination, not quantification. Several calibration procedures which reduce kinematic inaccuracies have been investigated ([19],[60],[70],[92]). All of these procedures utilize specialized measurement devices to account for manufacturer's fabrication tolerances. Those methods are well suited for industrial applications where the environment is well defined and the task is repetitive. In such environments uncertainty elimination techniques can be applied in the initial setup of the work cell to significantly decrease the uncertainty. Even with compensation for manufacturer's fabrication tolerances, calibration of the manipulator will still produce a degree of uncertainty dependent on the resolution of the joint angle measurement instrumentation used by the control system. No technique available for measuring uncertainty has been applied to the calibration problem.

There are two basic techniques for quantifing the level of uncertainty: bounding or probabilistic. In his research on robot planning Brooks treats measurement uncertainty by determining or assuming bounds on the measurement error [11]. An uncertainty is represented as:

$$- R \leq \text{measurement error} \leq S \qquad (2.1)$$

Where:

R = a lower error bound

S = an upper error bound

If the degree of measurement error is dependent on the measurement itself the uncertainty can be expressed as:

$$- R(m) \leq \text{measurement error} \leq S(m) \qquad (2.2)$$

Where:

m = the measured value

R(m) = A lower measurement dependent error bound

S(m) = An upper measurement dependent error bound

Therefore the true physical value, v, that a sensor reading of m represents is defined as:

$$m - R(m) \leq v \leq m + S(m) \qquad (2.3)$$

An illustrative example is presented in [11].

A problem with the bounded approach is the lack of information provided about the relative occurence of error values inside the bounded limits. For that reason uncertainty has usually been treated in a probabilistic manner. A distribution function is assigned to an event to represent the level of uncertainty involved in the occurence of that particular event being selected from a set of all possible events.

A brief review of the history of uncertainty and its probabilistic measure has been conducted by Saridis [85] and Sanderson [83]. The development of the Entropy function as a measure of uncertainty is clearly defined. The utilization of Entropy as a measure of uncertainty dates back as early as 1763. Since then, Entropy has played an important role in several fields, most notably, statistical thermodynamics and information theory.

If the probability of an individual random variable being selected from the set of available random variables can be expressed as:

$$P(X=x_i) = P(x_i) \geq 0 \quad (i=1...n) \tag{2.4}$$

$$\sum_{i=1}^{n} P(x_i) = 1 \tag{2.5}$$

Where:

X	= The random variable
x_i	= The discrete values of X
n	= Number of discrete values of X
$P(X=x_i)$	= Probability that X equals x_i

Then the Entropy of that probability distribution is given by:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i) \tag{2.6}$$

In the case of an event being defined by the occurence of more than one random variable the joint probability on the joint sample space is defined as:

$$P(X=x_i,\ Y=y_j) = P(x_i,\ y_j) \geq 0 \qquad (2.7)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m} P(x_i,\ y_j) = 1 \qquad (2.8)$$

Where:

X, Y = Random variables

x_i, y_j = Discrete values of X and Y

n, m = Number of discrete values of X and Y

$P(x_i,\ y_j)$ = Probability $X=x_i$ and $Y=y_j$.

Then the Entropy of that distribution is expressed as:

$$H(XY) = -\sum_{i=1}^{n} \sum_{j=1}^{m} P(x_i,\ y_j) \log_2 P(x_i,\ y_j) \qquad (2.9)$$

If occurance of the individual events is independent than the Entropy becomes:

$$H(XY) = -\sum_{i=1}^{n} \sum_{j=1}^{m} P(x_i)\ P(y_j) \log P(x_i)\ P(y_j) \qquad (2.10)$$

$$H(XY) = -\sum_{i=1}^{n} P(x_i) \log P(x_i) - \sum_{j=1}^{m} P(y_j) \log P(y_j) \qquad (2.11)$$

$$H(XY) = H(X) + H(Y) \qquad (2.12)$$

The additive property of the Entropy function [23] makes it an ideal performance measure for multileveled command structures operating in uncertain environments. Saridis has applied the concept of Entropy as a unifying performance function among the three levels of a hierarchical intelligent robotic system [85-6].

Sanderson has studied the utilization of Entropy as a common measure for evaluating the performance of part assembly system designs [83]. In that research he utilizes the concept of Entropy as a measure of the uncertainty in position and orientation of parts in an assembly task. Sanderson first defines parts Entropy for a one dimensional example analogous to equation 2.3. If the position probability is represented by a uniform distribution then the Entropy function is a maximum. Intuitively that makes sense since a uniform distribution provides the least amount of information on where the part is located and therefore produces the highest level of uncertainty.

Sanderson also demonstrates that knowledge of an estimate of part position produced by some measurement device reduces the part position uncertainty. The position probability can now be conditioned by the position estimate. The resultant conditional probability can be expressed as:

$$P(X=x_i|Y=y_j) = P(x_i|y_j) = P(x_i, y_j)/P(y_j) \qquad (2.13)$$

Where:

x_i represents the part position

$P(x_i, y_j)$ and $P(y_j)$ are as defined in eq. 2.4 and 2.8

$P(x_i|y_j)$ is the probability $X=x_i$ given that $Y=y_j$

y_j represents the estimated part position from a sensor reading.

The uncertainty of a part position conditioned on it's estimated position can be expressed by conditional Entropy defined as:

$$H(X|Y) = -\sum_{i-1}^{n} P(x_i|y_j) \log_2 P(x_i|y_j) \qquad (2.14)$$

Knowledge of estimated position doesn't eliminate the uncertainty but reduces it to a dependence on the range and resolution of the sensor system.

For a uniform distribution the probabilistic and bounded techniques for uncertainty measurement convey an equivalent amount of information. However only the Entropy measure can utilize new information to learn the distribution of the positions and therefore reduce the level of uncertainty on-line.

2.6 Summary

The original VAX/PUMA system is inadequate for continued real-time control research. Other real-time control systems operational in the fall of 1984 were not hierarchically based. Proposed reduction methods for Lagrange-Euler dynamical computations were insufficient. Newton-Euler reduction techniques had not been applied to a PUMA-600. Efficient computational forms of manipulator dynamics are essential for real-time implementation of proposed gross motion robotic control methods. The PUMA-600 has been characterized sufficiently to allow friction compensation and generation of achievable trajectories. Uncertainty quantification techniques have not been applied to manipulator calibration. Development of the theoretical basis for utilization of the Entropy function as a measure of calibration uncertainty would extend the unification of intelligent machine performance characterization to the lowest level.

The requirement for development of a hierarchical robotic evaluation environment and it's application to conduct a more complete and thorough real-time evaluations of modern control methods for large range robotic movement clearly exists. The development of such an environment is the subject of the next chapter.

CHAPTER 3

A HIERARCHICAL ROBOTIC EVALUATION ENVIRONMENT

3.1 Introduction

A major contribution of this research is an original solution to the problems that have constrained real-time evaluation of modern manipulator control techniques. That solution is a hierarchical robotic evaluation environment composed of three major integrated components: a hierarchical manipulator control system, customized efficient algorithms for computation of manipulator dynamics, and software libraries that support simulation and real-time modern control algorithm performance evaluation.

The principle of decreasing intelligence with increasing precision is applied in the design and implementation of a three stage hierarchical manipulator control system. A study of the PUMA dynamical formulations produces the necessary efficient dynamics algorithms. To support evaluation of modern control techniques, libraries of support software are developed. These libraries permit simulation evaluation of proposed algorithm effectiveness and allow the control system and custom dynamics to be integrated into a real-time robotic algorithm exerciser. A detailed summary of these components' development is presented in this chapter. Detailed documentation is provided in ([37-41],[44-5],[48])

3.2 Control System Development

The key component of any real-time robotic evaluation environment is the manipulator contıol system. Consequently the first integrated component to be developed was a hierarchically based PUMA control sy¢tem.

3.2.1 Motivation

Although the VAX/PUMA manipulator control system was utilized for the first real-time evaluation of four modern control techniques [103-4] it had several major limitations [48]:

1. When the VAX/PUMA system was installed, VAL was inoperative. In order to run VAL, cables and cards had to be disconnected and reconnected. That process was inconvenient and hard on the equipment.
2. The parallel data transfers via the DR11-W are inefficient without the development of custom device drivers. The minimum interface time for a read and write operation was 40ms.
3. The time shared nature of the VAX resulted in unpredictable sampling times and use of real-time priorities adversely affected other researchers productivity.
4. The VAX served as both the organizer and coordinator of the control system.

These limitations prevented the comprehensive evaluation of modern control techniques on the VAX/PUMA system.

3.2.2 A Hierarchical Computer Control System

In order to permit comprehensive robotic algorithm evaluation the VAX/PUMA control system was completely redesigned [36] under the following constraints:

1. usage of the existing operating system and languages on the RAL VAX,
2. switch selectable VAL controller, and
3. minimal additional hardware.

The switch selectable VAL constraint is imposed so that other PUMA related research could proceed concurrent with redesign of the original control system. The other constraints are due to economic considerations.

The control system redesign was heavily influenced by the original controller proposed by Valavanis [102], function guidelines for a robot controller suitable for inclusion in a manufacturing system proposed by researchers at the Carnegie-Mellon Robotics Institute [22] and experience with the VAX/PUMA system [50]. Due to the difficulties encountered with the DR11-W interface the decision was made to develop a loosely based hierarchy using serial links. That effort produced a Hierarchical Computer Control System (HCCS) for the PUMA-600 robot arm [36]. The HCCS hierarchy had three levels; organization, coordination and hardware control. Organizer responsibilities were handled by a VAX-11/750, which communicated over a serial link with the coordinator, a LSI-11/23 installed in place of the original LSI-11/02 computer. The hardware level remained unaltered. Now VAL could be operational when the HCCS was not. Selection of the LSI-11/23 as the coordinator computer allowed system development on a popular, well supported bus structure that is upward compatible to the growing family of PDP-11 products. PDP assembly language programs could also be written, compiled and tested on the VAX operating in MCR mode [106]. Software and hardware were sufficiently developed by January 1985 to permit Bang-Bang control of the PUMA-600 [36].

Employment of the HCCS for preliminary testing of simple real-time control algorithms exposed several major system limitations:

1. sampling rates were compatible with the VAX/PUMA system due to slow floating point execution,
2. insufficient memory,
3. limited serial interface speed, and
4. manipulator dependent software.

The hardware limitations have been overcome by the purchase of an FPF11 floating point processor, better use of the memory management unit and a DMF32 serial interface unit installed in the VAX. The original HCCS software was modified to be manipulator independent. Those modifications culminated in the creation of the RAL Hierarchical Control System (RHCS). A complete description of system hardware and software is contained in the RHCS user's guide [41]. An overview of the RHCS system is presented next.

3.2.3 The RAL Hierarchical Control System

The RHCS was designed to provided the following capabilities [41]:

1. the VAL controller is switch selectable,
2. the system primitives that control the manipulator form a library of modular subroutines callable from higher level languages,
3. the sample rate times can support real-time modern control evaluation,
4. the user interface is upward compatible with the VAX/PUMA system,
5. the system adheres to the principles of hierarchical control proposed by Saridis [86], and
6. coordinator and organizer level software and hardware are primarily manipulator independent.

The PUMA manipulator RHCS link block diagram is shown in figure 3.1. Under RHCS the control task is divided among the three levels of the hierarchy [41]. The organizer level is responsible for overall organization of the task and user interface through the power of the VAX. At the coordinator level organizer commands are translated into a series of control sequences to the manipulator hardware. The hardware level completes the desired command using manipulator dependent electronics. Only the hardware level abilities were resident in the original standalone robot controller. To empower the system with the abilities of a three stage hierarchy, libraries of organizer and coordinator software were developed.

A library of manipulator independent subroutines has been developed to implement the organizer level functions: downloading programs, interlevel communication, manipulator calibration control and system protection. The subroutines are written in VAX FORTRAN and make extensive use of VMS system calls. Table 3.1 lists these subroutines and their functions.

TABLE 3.1

RHCS ORGANIZER LEVEL SOFTWARE

DLOAD:	Download programs to the coordinator
PDPCOM:	Support interlevel communication
PDPINO:	Support interlevel general data transfer
OFRVAX:	Control recovery from range violation
PUMACAL:	Control PUMA manipulator calibration

The coordinator level functions: hardware control, interlevel communication, manipulator calibration and system protection are performed by a library of coordinator level subroutines. All coordinator subroutines are written in PDP assembly language for maximum speed advantage and have been developed on the VAX using the RSX11 VERSION 1.0 compilers

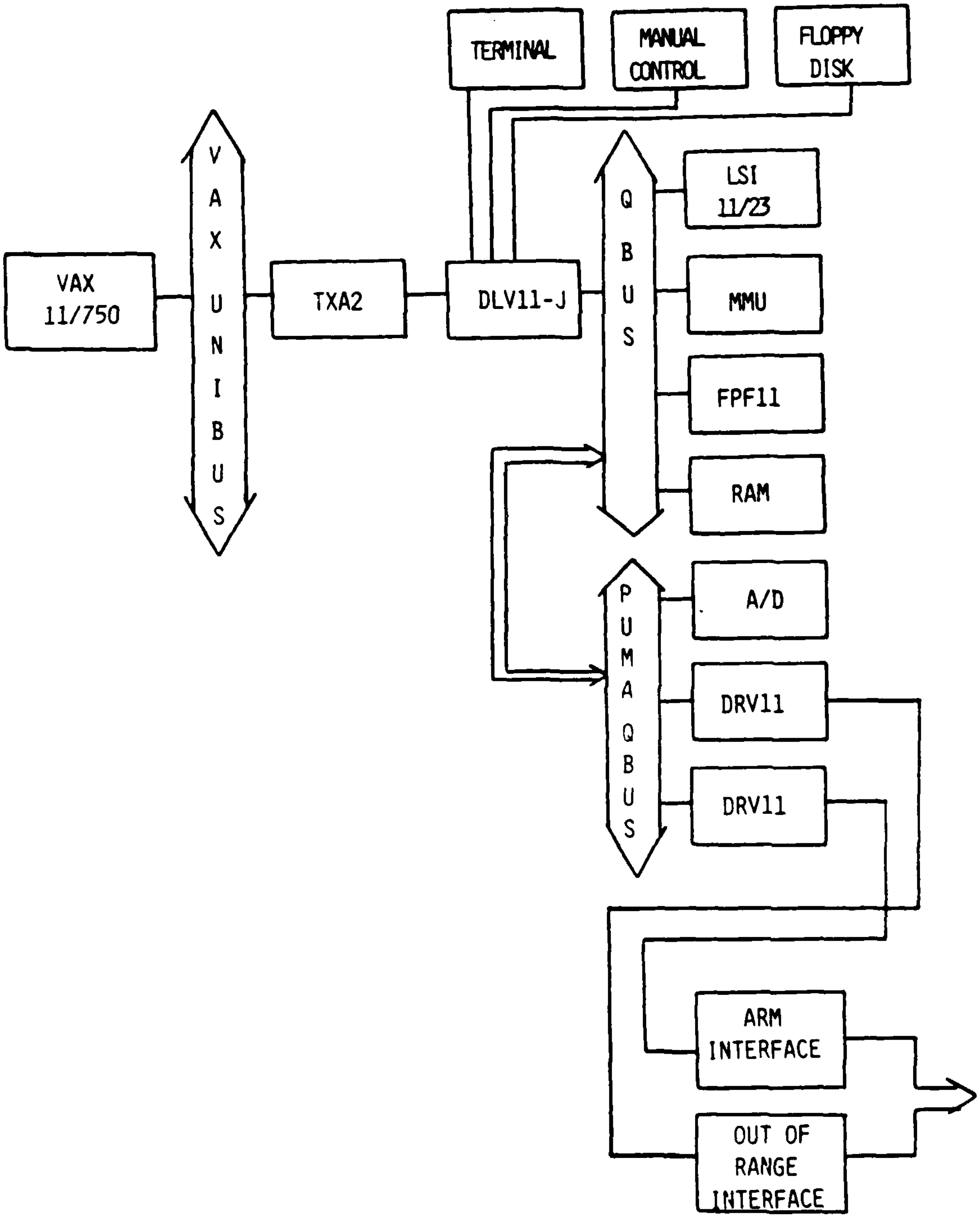


Figure 3.1 RAL Hierarchical Control System

[106]. Maximum utilization of the powerful LSI-11/23 addressing modes and the floating point processor has been accomplished. The assembly language subroutines have been created so that they can be called from higher level language programs. The new subroutines are similar to the ones written for the original system in both mnemonics and calling format. Only the coordinator level control software is manipulator dependent. Because of the proprietary nature of the arm interface, control programs are not available for general distribution without a legal release from Unimation. Tables 3.2-3 lists the RHCS coordinator level software and their functions.

By employment of a WCMODE, REPOS, ASTOP sequence the manipulator can be moved to any desired position without the restrictions of VAL's controller. Those three routines provide the control functions necessary to implement modern control algorithms under the RHCS. A flowchart of a general manipulator control law implementation employing the RHCS is displayed in figure 3.2.

To utilize the RHCS the user must write two programs, one for the organizer and another for the coordinator. As in the VAX/PUMA system, the user must call the PUMACAL subroutine before any attempt to load and execute an arm control algorithm. After calibration the organizer program generally ends up being a series of reads, writes, and PDPCOM calls. If a higher level language program is

TABLE 3.2

RHCS COORDINATOR LEVEL SOFTWARE
MANIPULATOR INDEPENDENT
SUBROUTINES and FUNCTIONS

COMMUNICATION	
RDVAX:	Read from organizer
SDVAX:	Send to organizer
PDPVAX:	Synchronized communication with organizer
INTERRUPT	
ETIMER:	Enable timer
DTIMER:	Disable timer
PROTECTION	
OFRPDP:	Detect range violations and stop arm

TABLE 3.3

RHCS COORDINATOR LEVEL SOFTWARE
MANIPULATOR INDEPENDENT FUNCTIONS
PUMA SUBROUTINES

CONTROL

BASIC

WRVECT:	Write vector to hardware level
WRSCLR:	Write scalar to hardware level
REVECT:	Read vector from hardware level
RESCLR:	Read scalar from hardware level
ENCANG:	Convert encoder count to joint angles

USER

WPMODE:	Write posmode command and data vector
WCMODE:	Write current mode command and data vector
WRCJNT:	Write current mode command and data
WCSTOP:	Stop an individual joint motion
ASTOP:	Stop motion of all joints
REPOS:	Read all joints angular position in degrees
RREPOS:	Read all joints angular position in radians
HPBOFF:	Enable mechanical brakes

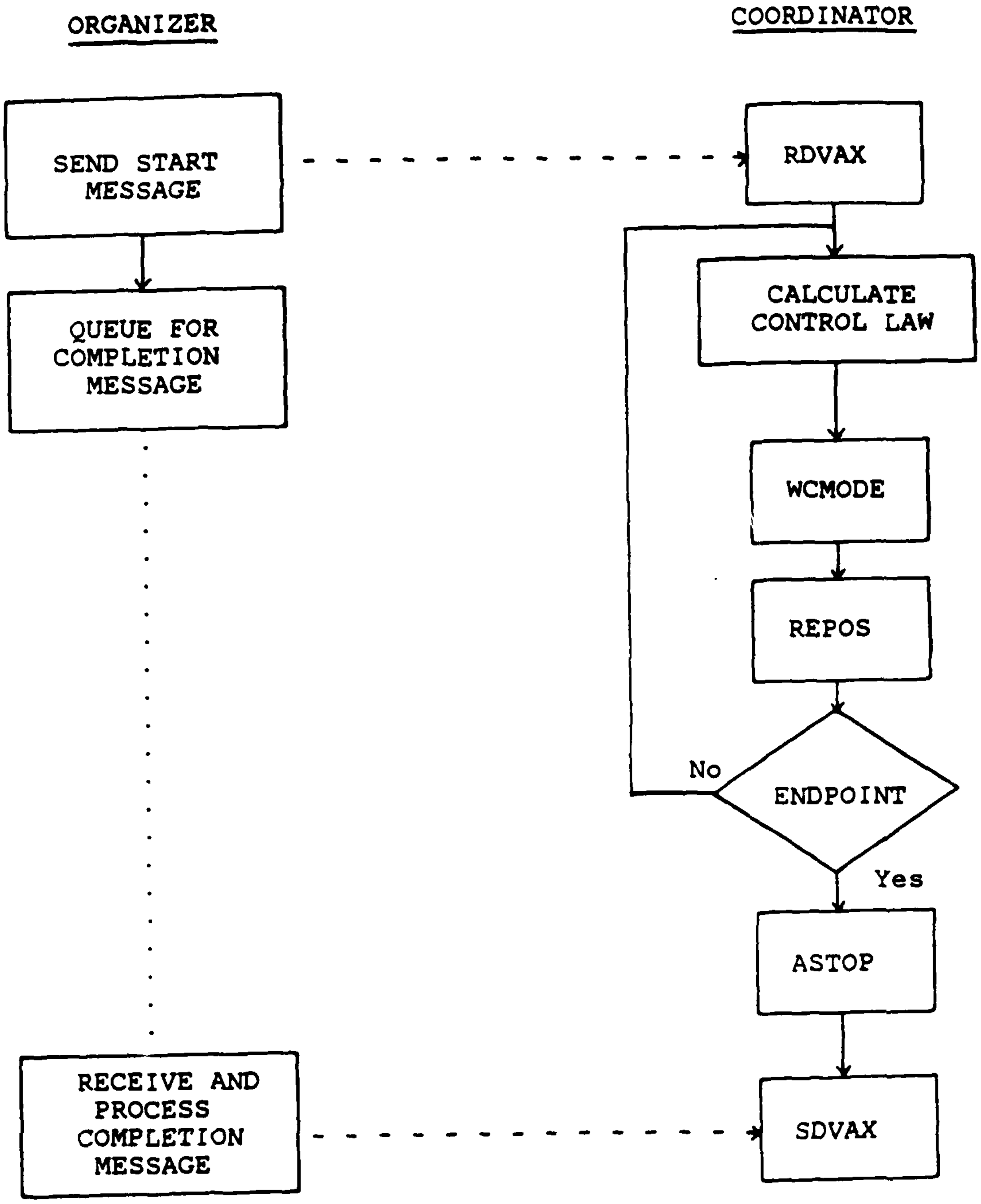


Figure 3.2 RHCS Control Flowchart

written for the coordinator level no system calls can be used and only the functions supported by an RSX11-M version of that language are allowed. All coordinator level programs upon completion must call COHEAD so that the system is ready to load and execute the next program.

Abstracts and calling formats for each organizer and coordinator subroutine are provided in the RHCS user's guide [41]. Listings of the organizer subroutines are contained in [37].

3.3 Efficient Dynamics Development

The RHCS supplies the tools necessary for controlling the PUMA without VAL in a hierarchical based intelligent work cell. Real-time evaluation of modern dynamics based robotic control methods presents the additional requirement of efficient computation of those dynamics. To fulfill that requirement custom Lagrange and Newton-Euler dynamics algorithms have been developed.

Efficient algorithms for the computation of Lagrange-Euler dynamics are realized through the use of REDUCE2 [27] and numerical significance analysis. The inertial matrix and gravity vector calculations developed in house [44] are more efficient than those proposed by Isaguiree and Paul [32] and the effects of the reductions are well documented.

The structure of the PUMA-600 is also used to significantly reduce the number of Newton-Euler computations [44]. Original work with real-time forms of the Newton-Euler equations ([51],[104]) has been expanded to encompass a set of reduction techniques similar to those suggested by Kanade [33] and Hollerbach [29]. The resultant set of custom PUMA-600 Newton-Euler equations are more efficient than the model proposed by Horak [30]. The first set of dynamic formulations sufficient to support 14ms sample rate real-time evaluation of dynamics based manipulator control techniques is now available.

Both the Lagrange and Newton-Euler customized algorithms have been coded in VAX FORTRAN for simulation use and in PDP assembly language for utilization in association with the RHCS. Table 3.4 lists these subroutines and their functions. Listings of all efficient dynamics algorithms are contained in [38].

3.4 Evaluation Environment Software Development

Comprehensive evaluation of modern robotic control methods requires an environment that supports both their simulation and real-time implementation. With the successful deployment of the RHCS, and creation of custom dynamics algorithms the foundation necessary to support comprehensive evaluation of modern robotic control techniques on a PUMA manipulator is firmly established.

TABLE 3.4

EFFICIENT DYNAMICS SOFTWARE SUBROUTINES and FUNCTIONS

ORGANIZER LEVEL (VAX FORTRAN)

RBTFLE:	Complete Lagrange-Euler dynamics
RBTCNE:	Complete Newton-Euler dynamics
CPDGCST:	Calculate LE dynamics constants

COORDINATOR LEVEL (PDP ASSEMBLY)

RBTMNE:	Complete Newton-Euler dynamics
FLEDG4:	Full Lagrange-Euler Inertial matrix, and gravity vector
DLEDG4:	Diagonal Lagrange-Euler Inertial matrix, and gravity vector
GDGCST:	Read and store LE dynamics constants

The environment necessary to support simulation and real-time implementation has been designed under the following criteria:

1. permit evaluation of any algorithm on any manipulator connected to the VAX via a RHCS link,
2. support testing of both joint and cartesian space control algorithms,
3. store test results in VAX files in a format suitable for graphical analysis,
4. allow a wide range of operational environments to be used,
5. allow user selectable sampling speeds in 7ms intervals,
6. allow user selectable loading configurations, and
7. be user friendly.

Libraries of modular organizer and coordinator level software have been developed to satisfy those criteria. Listings of the contents of those libraries are contained in ([37],[39]).

3.4.1 Organizer Level Software

To enable the interactive specification of operational configuration additional organizer level functions have been provided. Modular VAX FORTRAN subroutines permit selection of trajectories, sample rate, initial condition, manipulator loading, and storage and formatting of error data. Table 3.5 lists those subroutines and describes their functions.

TABLE 3.5

EVALUATION ENVIRONMENT ORGANIZER LEVEL SOFTWARE
SUBROUTINES and FUNCTIONS

GENERAL

SLCTIC:	Select initial condition
SLCTLD:	Select link6/load configuration
SLCTMN:	Select manipulator
SLCTTJ:	Select joint space trajectory
SLCTTX:	Select cartesian space trajectory

SIMULATION

SEOUT:	Store error data
SRGTST:	Test for range space violations

R3AGE

REOUT:	Read error data from the coordinator
ADOUT:	Read additional data from the coordinator

In simulation studies the evaluation is conducted strictly on the organizer level. Manipulator motion is simulated by a fourth order Runge-Kutta integration of the inverse efficient complete Lagrange-Euler PUMA dynamics ([38],[74]). Real-time evaluation involves a complex set of interactions between the organizer and coordinator levels. To eliminate that complexity and allow user friendly real-time evaluations, the RHCS, custom dynamics, and evaluation support software have been linked together to form R3AGE, The RAL Real-time Robotic Algorithm Exerciser. The R3AGE user's guide provides detailed utilization information [40].

3.4.2 The RAL Real-Time Robotic Algorithm Exerciser

The hierarchical principles embodied in the RHCS were expanded upon in designing R3AGE. Manipulators interface to the R3AGE environment through an RHCS communication link. The organizer and coordinator functions of the RHCS are the backbone of the algorithm exerciser. A flowchart of R3AGE organizer and coordinator level interaction is displayed in figure 3.3. Under the evaluation environment invoked by R3AGE, organizer and coordinator level interaction is transparent to the user. To support that interaction coordinator level subroutines which support; test configuration, error data output, custom dynamics, and trajectory point update have been developed. Table 3.6 lists those subroutines and their functions.

TABLE 3.6

R3AGE COORDINATOR LEVEL SOFTWARE SUBROUTINES and FUNCTIONS

MAGE: Interface joint space control algorithm to organizer level

MXAGE: Interface cartesian space control algorithm to organizer level

GETTRJ: Transfer joint space trajectory point data into user memory space

GETXRJ: Transfer cartesian space trajectory point data into user memory space

FRICTC: Compensate for manipulator friction

TESTST: Move manipulator into initial condition by position mode

EOVAX: Transfer position and velocity error data to organizer

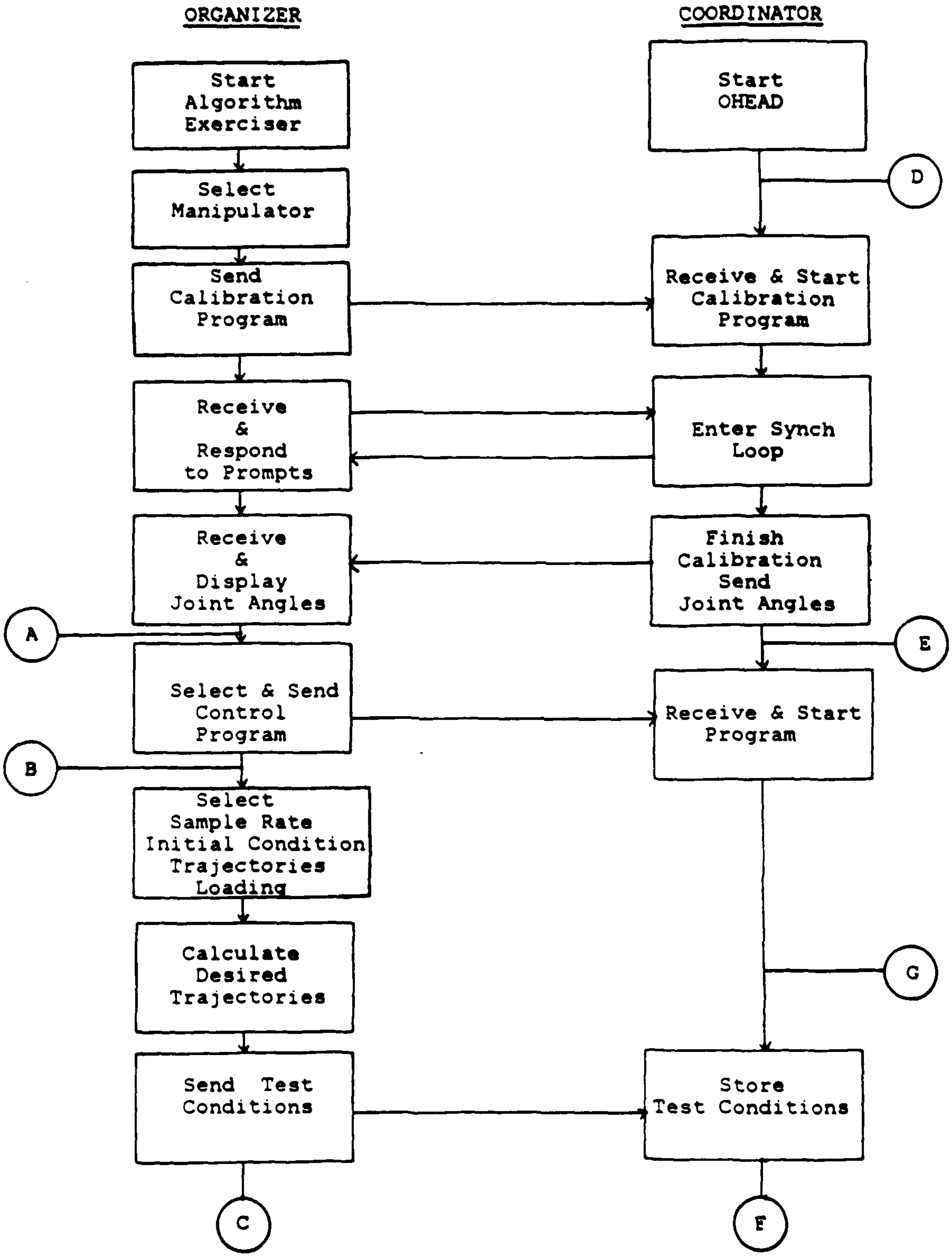

ORGANIZER
COORDINATOR
Start Algorithm Exerciser
Start OHEAD
D
Select Manipulator
Send Calibration Program
Receive & Start Calibration Program
Receive & Respond to Prompts
Enter Synch Loop
Receive & Display Joint Angles
Finish Calibration Send Joint Angles
A
E
Select & Send Control Program
Receive & Start Program
B
Select Sample Rate Initial Condition Trajectories Loading
Calculate Desired Trajectories
G
Send Test Conditions
Store Test Conditions
C
F

Figure 3.3a R3AGE Block Diagram page 1

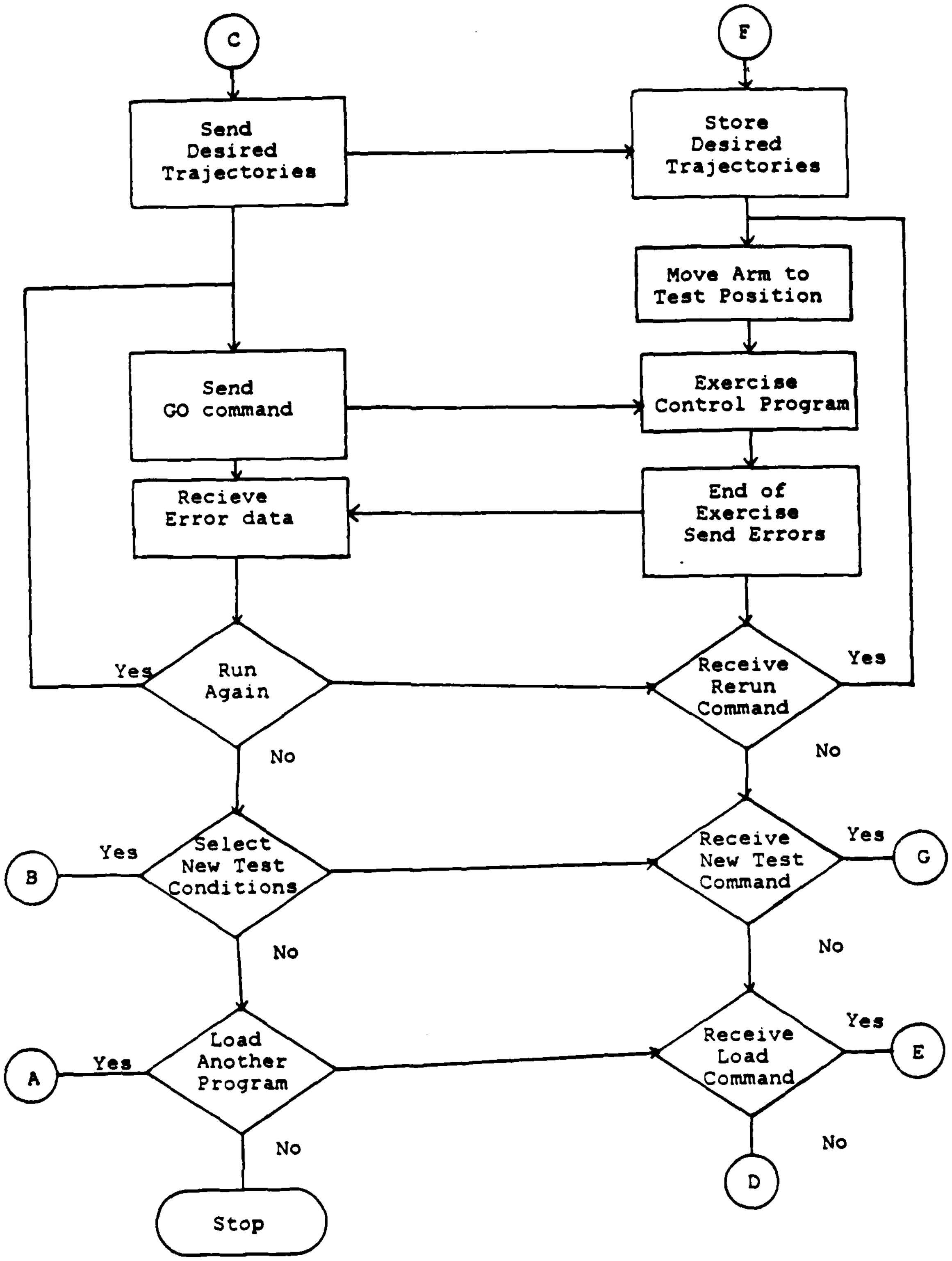


Figure 3.3b R3AGE Block Diagram page 2

R3AGE creates a user friendly environment that supports evaluation of manipulator control algorithms up to 27KB in length with a 4KB stack. Sufficient memory space for storage of 500 point position, velocity and acceleration trajectories, and error data, for a 6 degree of freedom manipulator is provided. R3AGE can be operated in interactive or automatic modes from a VAX terminal. In automatic mode the evaluation procedure is driven by commands stored in a file created by an automatic test file builder (BLDATF) ([37],[40]). Screen interaction under BLDATF duplicates that of R3AGE.

3.5 Summary

A three stage hierarchical control system for a PUMA manipulator has been developed. The RAL Hierarchical Control System (RHCS) provides the control primitives and communication protocol necessary for implementation of proposed theories for creation of hierarchically based intelligent work cells [86]. The control system software is upward compatible to the whole family of PDP systems. The RHCS organizer and coordinator levels are manipulator independent. Modifications to hardware level commands permit multiple non-identical manipulators to be networked by a series of RHCS links.

Symbolic reduction and significance analysis has produced a series of dynamics algorithms with computational efficiency sufficient to support RHCS implementation of dynamics based modern control techniques. Libraries of software that support simulation and real-time evaluation of modern control techniques have been developed.

An original solution to the problems constraining real-time evaluation of modern robotic control techniques was created by integration of the RHCS, efficient dynamics, and evaluation support software, into the RAL Real-Time Robotic Algorithm Exerciser (R3AGE). The environment created under R3AGE supported the real-time evaluation of dynamics for robotic control presented in the next chapter.

CHAPTER 4

EVALUATION OF DYNAMICS FOR ROBOT CONTROL

4.1 Introduction

A large class of manipulator control techniques utilize some form of dynamical modeling in their control laws [56]. Those techniques assume a degree of modeling accuracy sufficient for cancellation of the effects of the actual dynamics by the mathematical expression of the manipulator dynamics contained in the control law. If that assumption is valid, any tracking errors may be asymptotically driven to zero. Therefore, knowledge of the effect of neglected dynamics on control law effectiveness would be an invaluable aid in real-time implementation of modern robotic control techniques.

The computed-torque technique is the most basic representation of the dynamically dependent control philosophy. The heuristic global linearization scheme of the computed-torque technique produces a control law analogous to the mathematically based exact linearization [95], nonlinear feedback [20], and optimal control methods [65]. Knowledge about the effect of dynamics on robot control can be obtained from evaluation of the performance ramifications produced by varying the manipulator representation contained in the computed-torque control law.

In this chapter a significant contribution in the area of manipulator control research is made by the evaluation of dynamics models for simulated and real-time control of a six degree of freedom PUMA manipulator. Those evaluations form a manipulator control performance database. The effects of inertial coupling, Coriolis and centrifugal forces, and actuator inertias are identified by analysis of their impact on the accuracy of the computed-torque control algorithm. Different feedforward loop manipulator dynamical models produce the best simulated and real-time controller performance. Complete dynamics in the feedforward loop produces the optimum simulation performance. Simulation tracking accuracy degrades as a function of model incompleteness.

Utilization of uncoupled dynamics in the feedforward loop produces the best overall real-time control algorithm performance. Real-time tracking accuracy degrades as a function of model completeness. Forces not modeled by a Lagrange-Euler formulation dominate the real dynamics of the PUMA manipulator. A new representation of the real PUMA dynamics is identified for realistic simulation of modern control algorithms and improved real-time performance.

4.2 Method Of Approach

The computed-torque technique ([55],[78]) provides a mathematically well defined, dynamically dependent basic control algorithm for the study of the effects of dynamics on real-time robotic control. The dynamical formulations employed in the controller are: complete Newton-Euler, and three reduced forms of Lagrange-Euler: full, block, and diagonal inertia matrix, all with gravity but without Coriolis and centrifugal terms. By evaluating PUMA manipulator performance variations the effects of computed-torque feedforward loop neglected dynamics on gross motion joint control are exposed.

To obtain comprehensive information about the effects of dynamics on algorithm performance the four algorithms have been evaluated over six different operational environments. The six test configurations can be subdivided into two blocks:

1. slow trajectory unloaded, and
2. fast trajectory unloaded.

Each block consists of three separate trajectories with identical velocity and acceleration profiles but different initial positions. The three sets of initial conditions (IC0,IC1,IC2) are displayed in table 4.1 along with a data key. The fast trajectories shown in figure 4.1 are derived from a performance characterization study of the PUMA arm [43]. The actual position trajectory is the sum of the incremental base trajectory and the selected initial position [40]. The peak velocity of each joint is achieved while avoiding real-time acceleration and torque saturation effects [43]. The slow trajectory has identical final positions but

TABLE 4.1a CHAPTER 4 DATA KEY

TITLE = XCTISMT

X - Test type.

- S - Simulation with actuators inertias
- N - Real-time with actuator inertias

CT - Control algorithm identifer.

- 10 - Newton-Euler dynamics
- 12 - Diagonal inertia dynamics
- 13 - Full inertia dynamics
- 14 - Block inertia dynamics

I - Initial condition specifier

- 0 - IC0 (0,-90,90,0,1,0,)
- 1 - IC1 (0,-135,135,0,1,0)
- 2 - IC2 (90,0,0,90,90,90)

S - Trajectory speed specifier

- 0 - Slow speed
- 1 - Fast speed

M - External load specifier

- 0 - unloaded
- 1 - fully loaded(2.3kg)

T - Sampling time specifier

- 1 - 7ms
- 2 - 14ms
- 3 - 21ms

TABLE 4.1b CHAPTER 4 SYMBOL KEY

Figure 4.2 Symbol Key

○	S101102
□	S101103

Figure 4.3 and 4.4 Symbol Key

□	X101103
○	X121102
▶	X131102
◇	X141102

FAST BASE TRAJECTORIES

SYMBOL KEY

⊡ Joint 1

⊙ Joint 2

▶ Joint 3

◇ Joints 4-6

VELOCITY

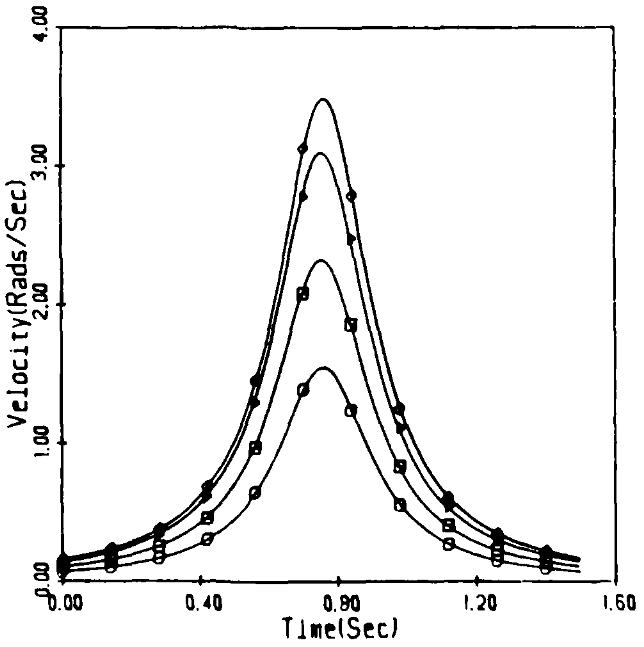
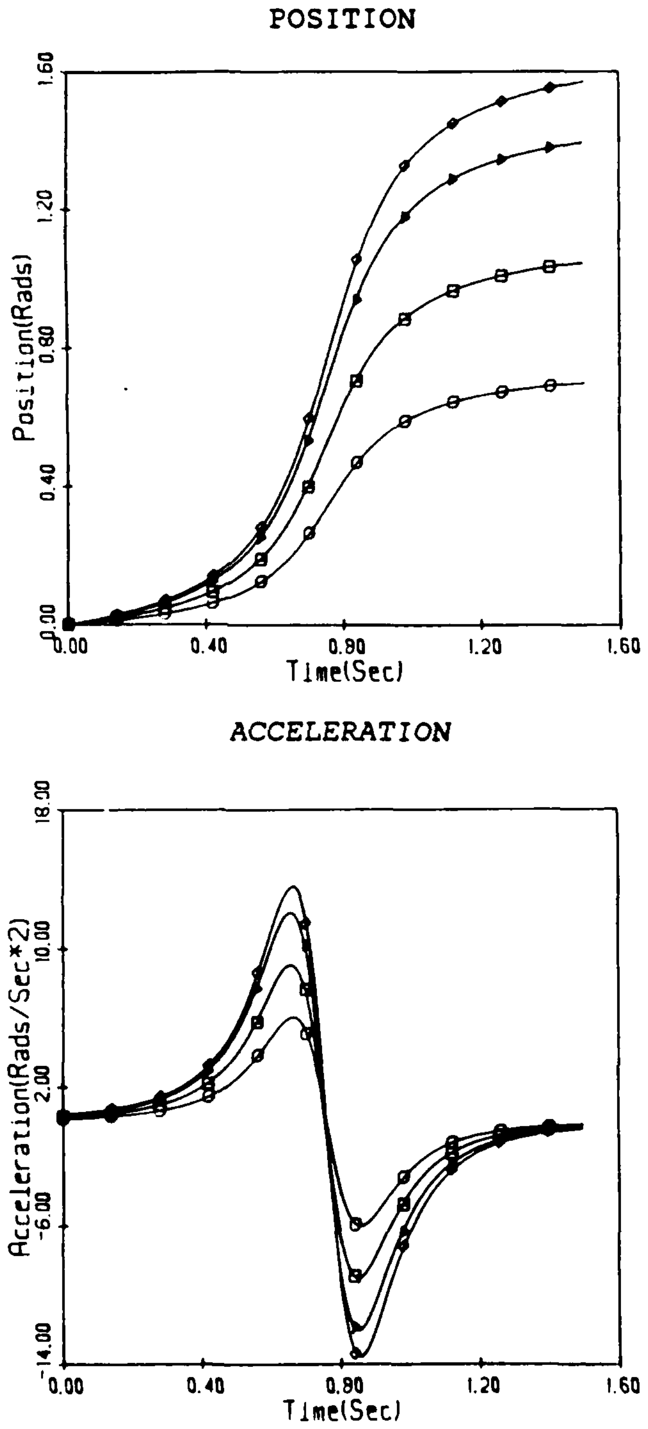


Figure 4.1a Fast Velocity Trajectories and Symbol Key

FAST BASE TRAJECTORIES



POSITION

ACCELERATION

Figure 4.1b Fast Position and Acceleration Trajectories

reduced velocities and accelerations due to a 25 percent elongation of the trajectory time. Joint 3 trajectories are reversed so that links 2 and 3 rotate in the same direction. For initial condition 2 (IC2) the joint trajectories are reversed to evaluate motion against gravity.

The four computed-torque algorithms are simulated on a VAX-11/750 using a fourth order Runge-Kutta integration routine with a one millisecond step size. The PUMA manipulator is simulated by a complete Lagrange-Euler dynamic model. Inertial arm parameters have been obtained from [44]. Actuator inertia magnitudes were derived in [94]. Real-time control algorithm evaluation is accomplished through utilization of the RAL Real-Time Robotic Algorithm Exerciser, R3AGE ([40],[48]). A 14ms sampling rate is selected for all algorithms except the Newton-Euler formulation which requires 21ms for real-time implementation.

To quantitatively compare the effects of dynamic models on robot control algorithm performance the power ranking formula shown in table 4.2 is employed. Controller performance is compared in four categories; peak and final, position and velocity errors. The normalized absolute error values in each category are weighted and summed to produce a relative indication of algorithm performance. The algorithm with the best performance will display the highest ranking. Power rankings range frcm zero to ten. A rank of zero indicates maximum error in all four categories. Power rank

TABLE 4.2

POWER RANK FORMULATION

$$PR_{CTi} = [1.0 - (SC_1 \times NPP_{CTi} + SC_2 \times NFP_{CTi} + SC_3 \times NPV_{CTi} + SC_4 \times NFV_{CTi})] \times 10$$

$$NPP_{CTi} = PP_{CTi}/MAX(PP_{ji})$$

$$NPV_{CTi} = PV_{CTi}/MAX(PV_{ji})$$

$$NFP_{CTi} = FP_{CTi}/MAX(FP_{ji})$$

$$NFV_{CTi} = FV_{CTi}/MAX(FV_{ji})$$

Where:

CT = Control algorithm identifier

j = Control algorithms (10, 12, 13, 14)

i = Joint identifier (1-6)

PR = Power ranking

PP = Absolute peak position error

PV = Absolute peak velocity error

FP = Absolute final position error

FV = Absolute final velocity error

SC = Scale factor $\sum_{i=1}^{n} SC_i = 1.0$

$SC_1 = 0.3$
$SC_2 = 0.3$
$SC_3 = 0.2$
$SC_4 = 0.2$

differences greater than three illustrate large variations in at least one category.

Evaluations are conducted with and without modeling actuator inertias. The effects of inertial coupling and Coriolis and centrifugal forces on simulated and real-time algorithm accuracy are analyzed for the included actuator case. Variations in those observations resultant from actuator modeling are then presented.

4.3 Computed-torque Technique Dynamic Models

The computed-torque technique employs both feedforward and feedback elements to control a robot arm ([55],[78]) and is a special case of the optimal control law [65]. The feedforward component uses manipulator dynamics to compensate for nonlinearities and coupling among the six joints. The feedback component computes necessary corrective torques to compensate for any deviations from the desired trajectory.

The computed-torque control law is:

$$\tau(t)=D(q)[\ddot{q}_d(t)+K_v(\dot{q}_d(t)-\dot{q}(t))+K_p(q_d(t)-q(t))]+h(q,\dot{q})+g(q) \qquad (4.1)$$

Where:

$\tau(t)$ = Vector of joint torques

q_d, $\dot{q}_d$, $\ddot{q}_d$ = Vectors of desired position, velocity, and acceleration in generalized joint coordinates

q, $\dot{q}$, $\ddot{q}$ = Vectors of measured position, velocity and acceleration in generalized joint coordinates

$D(q)$ = n x n inertial matrix

K_v = n x n derivative feedback gain matrix

K_p = n x n position feedback gain matrix

$h(q,\dot{q})$ = Vector of Coriolis and centrifugal forces

$g(q)$ = Vector of gravity forces

The Lagrange-Euler equation of motion for a manipulator is:

$$\tau(t) = \overline{D}(q)\ddot{q}(t) + \overline{h}(q,\dot{q}) + \overline{g}(q) \qquad (4.2)$$

where the overscore signifies actual values. Substituting equation 4.1 into equation 4.2 produces:

$$\overline{D}(q)\ddot{q}(t)+\overline{h}(q,\dot{q})+\overline{g}(q) = D(q)[\ddot{q}_d(t)+K_v(\dot{q}_d(t)-\dot{q}(t)) + K_p(q_d(t)-q(t))]+h(q,\dot{q})+g(q) \qquad (4.3)$$

If the modeled and actual dynamics are equal equation 4.3 reduces to:

$$D(q)[\ddot{e}(t)+K_v\dot{e}(t)+K_pe(t)] = 0 \qquad (4.4)$$

Where:

$$e(t) = q_d(t)-q(t)$$

Proper selection of feedback gains produces characteristic roots of equation 4.4 with negative real values driving the error to zero asymptotically. In order to obtain a critically damped system for each joint subsystem the corresponding elements in the diagonal feedback gain matrices obey the relationship, Kv=2sqrt(Kp). In this study the velocity and position gain matrices are equal for each joint and have values of 20 and 100 respectively placing the system poles at -10. Linear quadratic design techniques can be employed to obtain a set of optimal gain matrices [65].

Dynamics based control laws can be implemented with either Lagrange or Newton-Euler dynamics. A block diagram of the computed-torque control law utilizing Newton-Euler dynamics is shown in figure 4.1c. The control law that diagram represents is obtained by substituting:

$$\ddot{q}(t) = \ddot{q}_d(t) + K_v(\dot{q}_d(t)-\dot{q}(t))+K_p(q_d(t)-q(t)) \tag{4.5}$$

into the Newton-Euler dynamical equations.

A block diagram of the computed-torque control law utilizing Lagrange-Euler dynamics without Coriolis and centrifugal feedforward compensation is shown in figure 4.1d. The control law is now:

$$\tau(t) = D(q)[\ddot{q}_d(t)+K_v(\dot{q}_d(t)-\dot{q}(t))+K_p(q_d(t)-q(t))]+g(q) \tag{4.6}$$

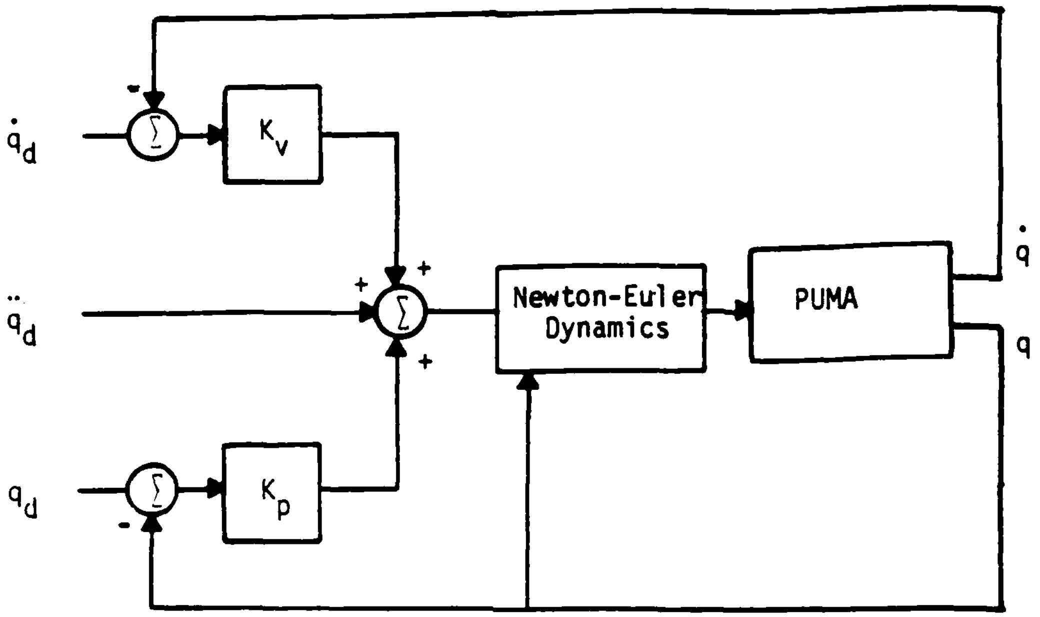


Figure 4.1c Newton-Euler Feedforward Dynamics Computed-Torque Block Diagram

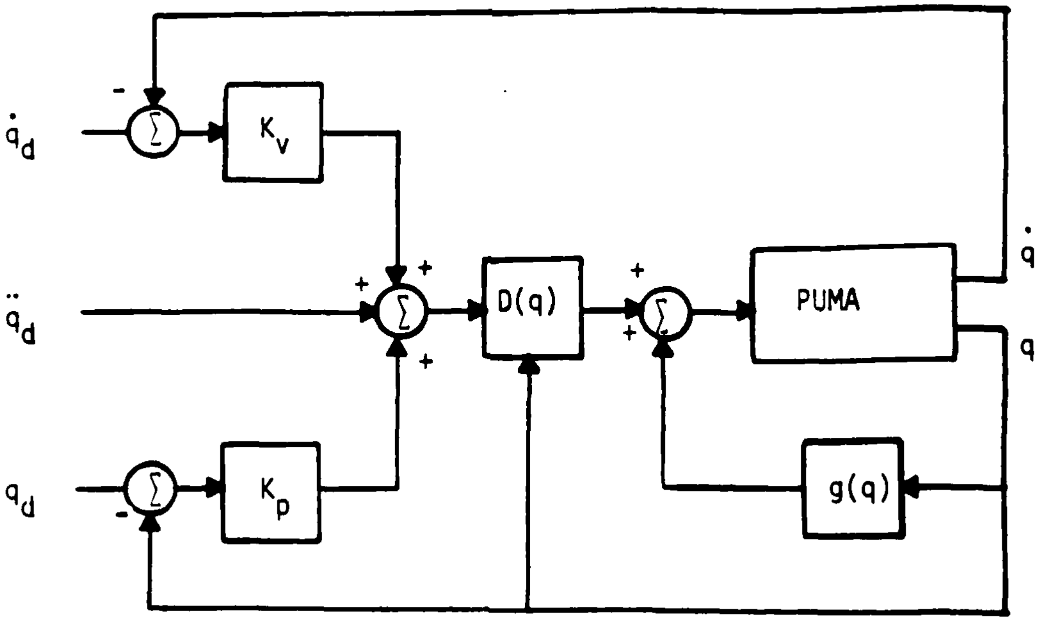


Figure 4.1d Lagrange-Euler without Coriolis and Centrifugal Feedforward Dynamics Computed-Torque Block Diagram

For these evaluations four forms of dynamics ([56],[77],[84]) have been employed in the feedforward loop:

1. Complete, using a Newton-Euler formulation.
2. Lagrange-Euler full inertia matrix without Coriolis and centrifugal terms.
3. Lagrange-Euler block inertia matrix without Coriolis and centrifugal terms with block inertia defined as:

$$\mathrm{blKD}(q) = \begin{bmatrix} D_{11} & D_{12} & D_{13} & 0 & 0 & 0 \\ D_{21} & D_{22} & D_{23} & 0 & 0 & 0 \\ D_{31} & D_{32} & D_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & D_{44} & D_{45} & D_{46} \\ 0 & 0 & 0 & D_{54} & D_{55} & D_{56} \\ 0 & 0 & 0 & D_{64} & D_{65} & D_{66} \end{bmatrix}$$

4. Lagrange-Euler diagonal inertia matrix without Coriolis and centrifugal terms.

4.4 Dynamic Model Simulation Evaluation

Simulation studies revealed that the rank order of algorithm performance is unaffected by trajectory speed. The essential conclusions about the effects of dynamic models on robot control algorithm performance are extracted from analysis of controller effectiveness in tracking the fast trajectory starting from different initial conditions. By employing the various starting points, the masking of important trends by gravity and other position dependent forces is avoided.

For links 2, 3 and 6 the basic shape of the error profiles is independent of the initial conditions. Joint 1 error patterns exhibit no correlation between initial conditions. For the highest ranking algorithm at any initial condition, the degree of dominance becomes more significant with increased load [49]. The best overall performance has been obtained by utilizing the full inertial dynamic model.

The comparison of fast trajectory IC1 individual position and velocity errors illustrated in figures 4.2-4 is included as a worst case representation of error profiles. Tables 4.3-5 present fast block power ranking comparison. Additional data figures and tables are included in [49]. Listings of the computed-torque algorithms employed in the simulation evaluation are contained in [39].

4.4.1 Effects Of Inertial Coupling

Inertial coupling between large and small joints has a minimal impact on large link control accuracy. Repercussions from neglecting large link coupling in small link control are significant for joints 4 and 5. Lack of knowledge about large link motion produces alterations in both link 4 and 5 magnitudes and pattern of the velocity and position errors. Link 6 effectiveness is not degraded when large link coupling is ignored in the feedforward loop.

Inertial coupling among the large links is a dominant factor in large link control. Neglecting that coupling causes excessive trajectory overshoot prior to the midpoint and lag thereafter. The resultant peak position errors are five to ten times larger than for links 2 and 3. Aligning joints 2 and 3 produces the maximum impact on link 1 from neglected coupling.

Variations in link 4 and 5 control algorithm performance attributed to ignoring inertial coupling among the small links is minimal. Lack of small link coupling has a negligible impact on link 6 error profiles.

A study of open-loop inertial coupling effects demonstrated that large link coupling is the dominant component in small link inertial torque composition and that the small links exhibit minimal coupling among themselves [44]. Closed-loop observations reinforce those conclusions.

Therefore an inertial matrix defined as:

$$\left[\begin{array}{ccc|ccc} D_{11} & D_{12} & D_{13} & 0 & 0 & 0 \\ D_{21} & D_{22} & D_{23} & 0 & 0 & 0 \\ D_{31} & D_{32} & D_{33} & 0 & 0 & 0 \\ \hline D_{41} & D_{42} & D_{43} & D_{44} & 0 & 0 \\ D_{51} & D_{52} & D_{53} & 0 & D_{55} & 0 \\ D_{61} & D_{62} & D_{63} & 0 & 0 & D_{66} \end{array}\right]$$

should represent all the necessary inertial coupling information while reducing control law computations.

4.4.2 Effects Of Coriolis And Centrifugal Forces

The repercussions from ignoring nonlinear dynamics are deduced by comparing simulated performance of the Newton-Euler and full inertial algorithms. Large joint error profiles are similar but peak position error magnitude is dependent on initial condition. Small joint error profiles are similar to those produced from utilization of the full inertial feedforward loop. At certain initial conditions, degradation from the increased sample period required by the complete dynamics offsets enhancements due to model accuracy for all links except the fifth. Inclusion of nonlinear forces in joint 4 dynamics severely degrades tracking ability from two of the three initial conditions independent of sample period.

Table 4.3 illustrates the impact on algorithm simulation power ranking from increasing the Newton-Euler dynamics sampling period from 14 to 21 ms. Invariance in full, block and diagonal power ranking demonstrates that one of those algorithms still produces the maximum error in all four ranking categories. A change in power ranking for all algorithms indicates that the complete model now produces the maximum error in at least one category. The errors variations due to extending the sample period are concentrated at the peaks with a maximum increase of 50 percent as illustrated in figure 4.2.

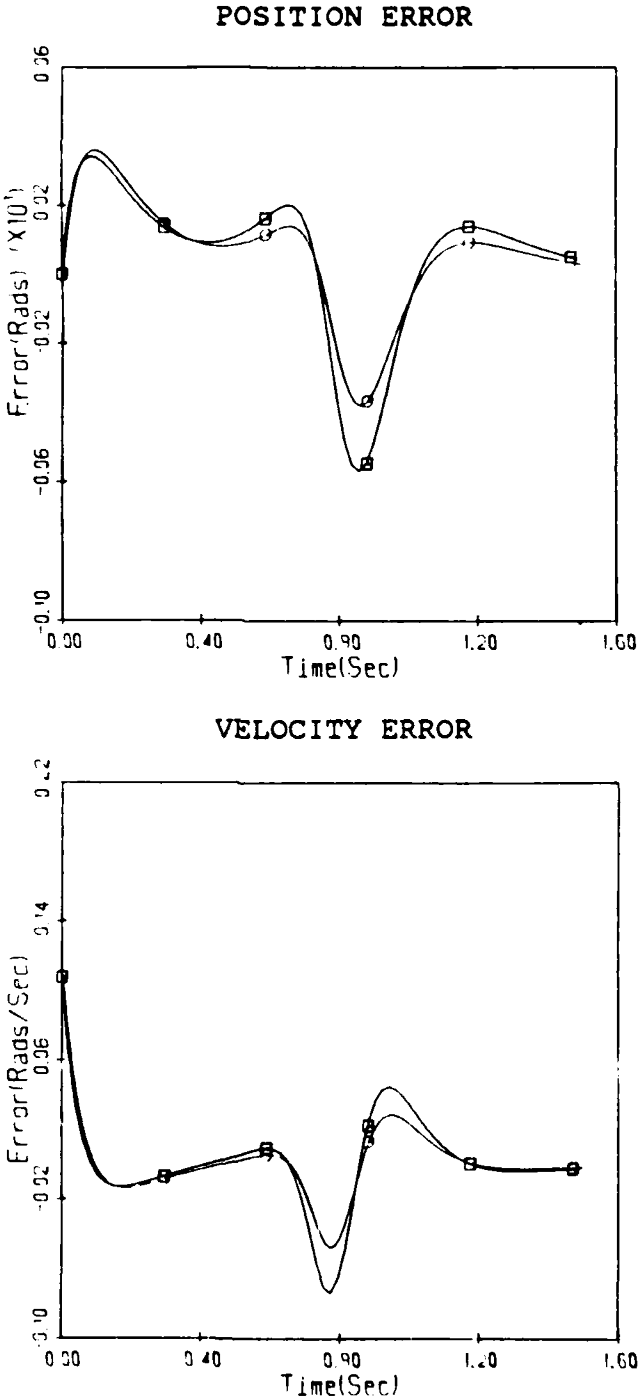
TABLE 4.3

ALGORITHM SIMULATION POWER RANKING
VARIATIONS DUE TO INCREASED NEWTON-EULER SAMPLING PERIOD
FAST TRAJECTORY

	21ms				14 ms			
DYNAMICS	Newton Euler	Full inertial	Block inertial	Diagonal inertial	Newton Euler	Full inertial	Block inertial	Diagonal inertial
IC0								
JOINT 1	5.96	5.54	5.63	0.00	7.29	5.54	5.63	0.00
JOINT 2	8.69	9.17	9.19	0.00	9.13	9.17	9.19	0.00
JOINT 3	8.05	8.09	8.54	0.00	8.70	8.09	8.54	0.00
JOINT 4	3.36	6.30	1.76	1.30	4.52	5.53	0.63	0.20
JOINT 5	1.51	1.80	2.23	1.00	4.04	1.35	1.85	0.60
JOINT 6	0.00	3.34	3.26	3.31	0.24	0.28	0.16	0.23
IC1								
JOINT 1	4.80	2.35	2.55	1.51	6.54	2.35	2.55	1.51
JOINT 2	8.79	8.97	8.94	0.00	9.20	8.97	8.94	0.00
JOINT 3	8.49	8.77	8.92	0.00	9.00	8.77	8.92	0.00
JOINT 4	1.70	4.96	2.46	2.62	2.73	3.61	0.88	1.14
JOINT 5	4.40	4.16	0.14	0.28	6.29	4.16	0.14	0.28
JOINT 6	0.00	3.34	3.27	3.34	0.24	0.27	0.16	0.26
IC2								
JOINT 1	0.03	6.19	5.86	2.87	1.51	5.82	5.50	1.40
JOINT 2	8.87	9.06	9.10	0.00	9.25	9.06	9.10	0.00
JOINT 3	7.95	4.00	4.32	0.01	8.64	4.00	4.32	0.01
JOINT 4	2.47	5.10	1.94	1.67	3.90	4.29	1.30	0.95
JOINT 5	3.53	5.46	1.72	1.59	5.01	4.92	1.24	1.01
JOINT 6	0.00	3.31	3.44	3.76	0.11	0.07	0.25	0.71

Power rankings illustrate relative performance by scaling and summation of the normalized peak and final position and velocity errors produced by different algorithms over identical trajectories. Power rankings range from zero to ten with the best performing algorithm annotated by the highest ranking. For additional information refer to table 4.2

SAMPLING RATE PERFORMANCE COMPARISON
COMPLETE DYNAMICS



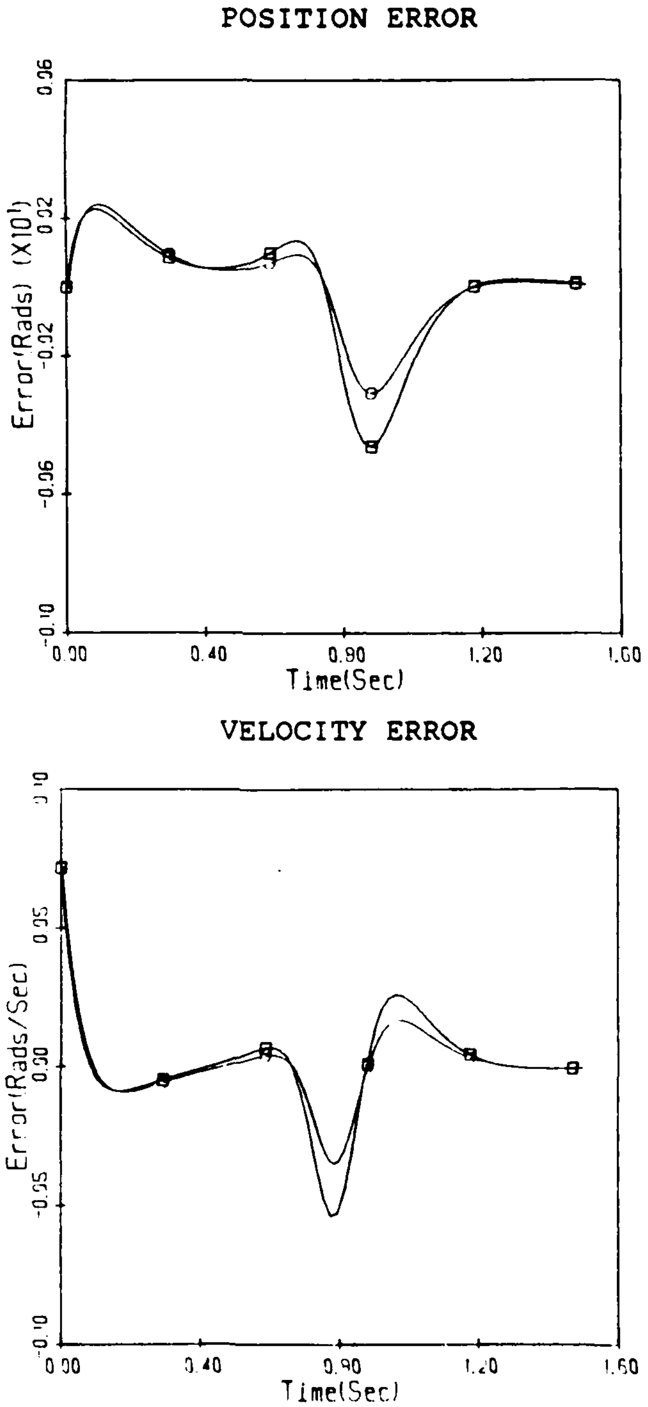
POSITION ERROR

VELOCITY ERROR

Refer to Table 4.1b for symbol definition

Figure 4.2a Joint 1 Fast IC1 Sample Rate Comparison

SAMPLING RATE PERFORMANCE COMPARISON
COMPLETE DYNAMICS



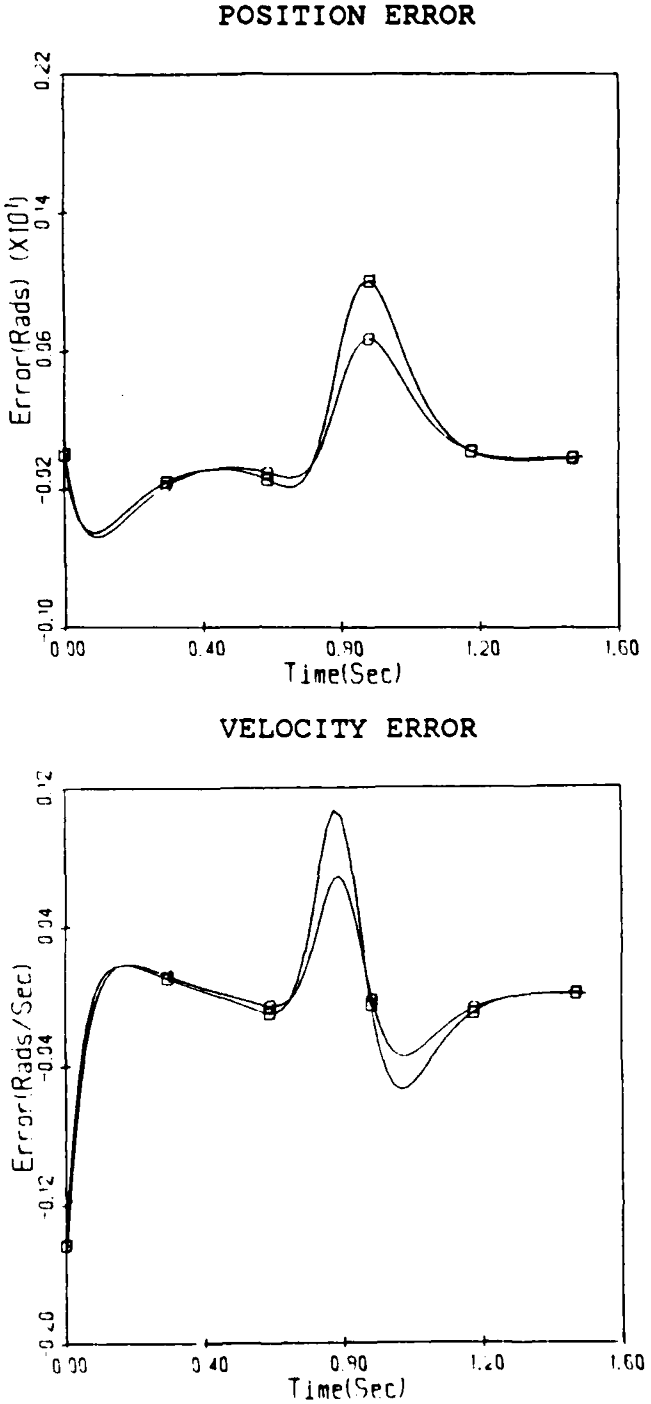
POSITION ERROR

VELOCITY ERROR

Refer to Table 4.1b for symbol definition

Figure 4.2b Joint 2 Fast IC1 Sample Rate Comparison

SAMPLING RATE PERFORMANCE COMPARISON
COMPLETE DYNAMICS



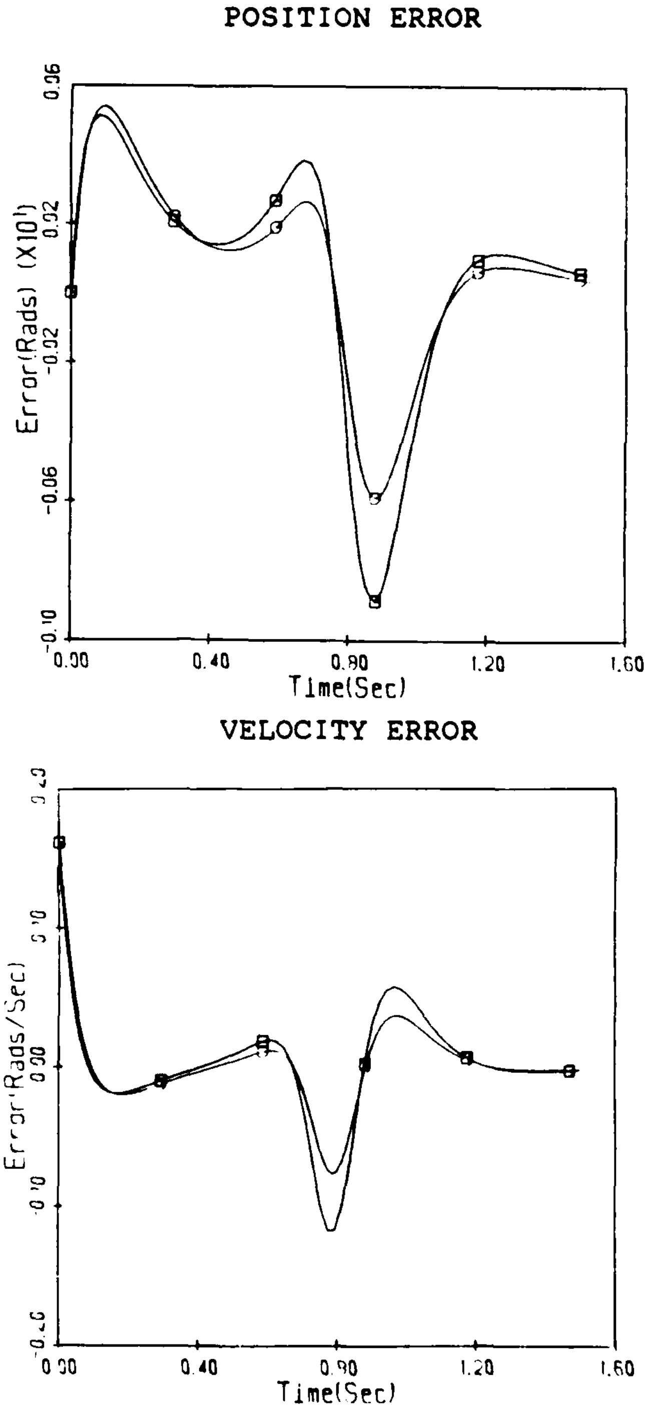
POSITION ERROR

VELOCITY ERROR

Refer to Table 4.1b for symbol definition

Figure 4.2c Joint 3 Fast IC1 Sample Rate Comparison

SAMPLING RATE PERFORMANCE COMPARISON
COMPLETE DYNAMICS

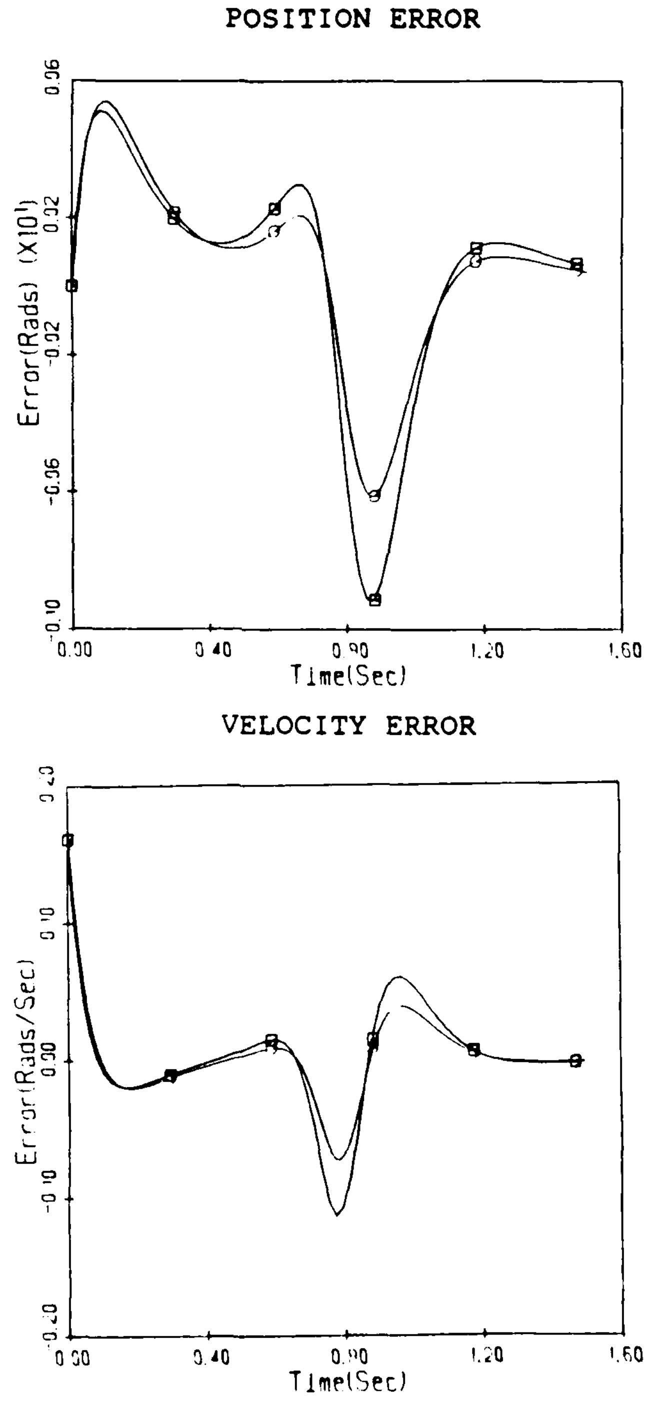

POSITION ERROR

VELOCITY ERROR

Refer to Table 4.1b for symbol definition

Figure 4.2d Joint 4 Fast IC1 Sample Rate Comparison

SAMPLING RATE PERFORMANCE COMPARISON
COMPLETE DYNAMICS


POSITION ERROR

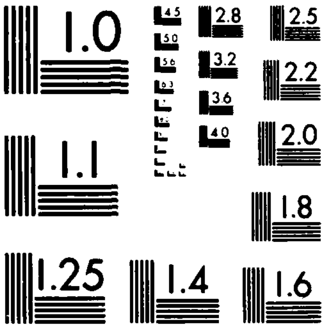
VELOCITY ERROR

Refer to Table 4.1b for symbol definition

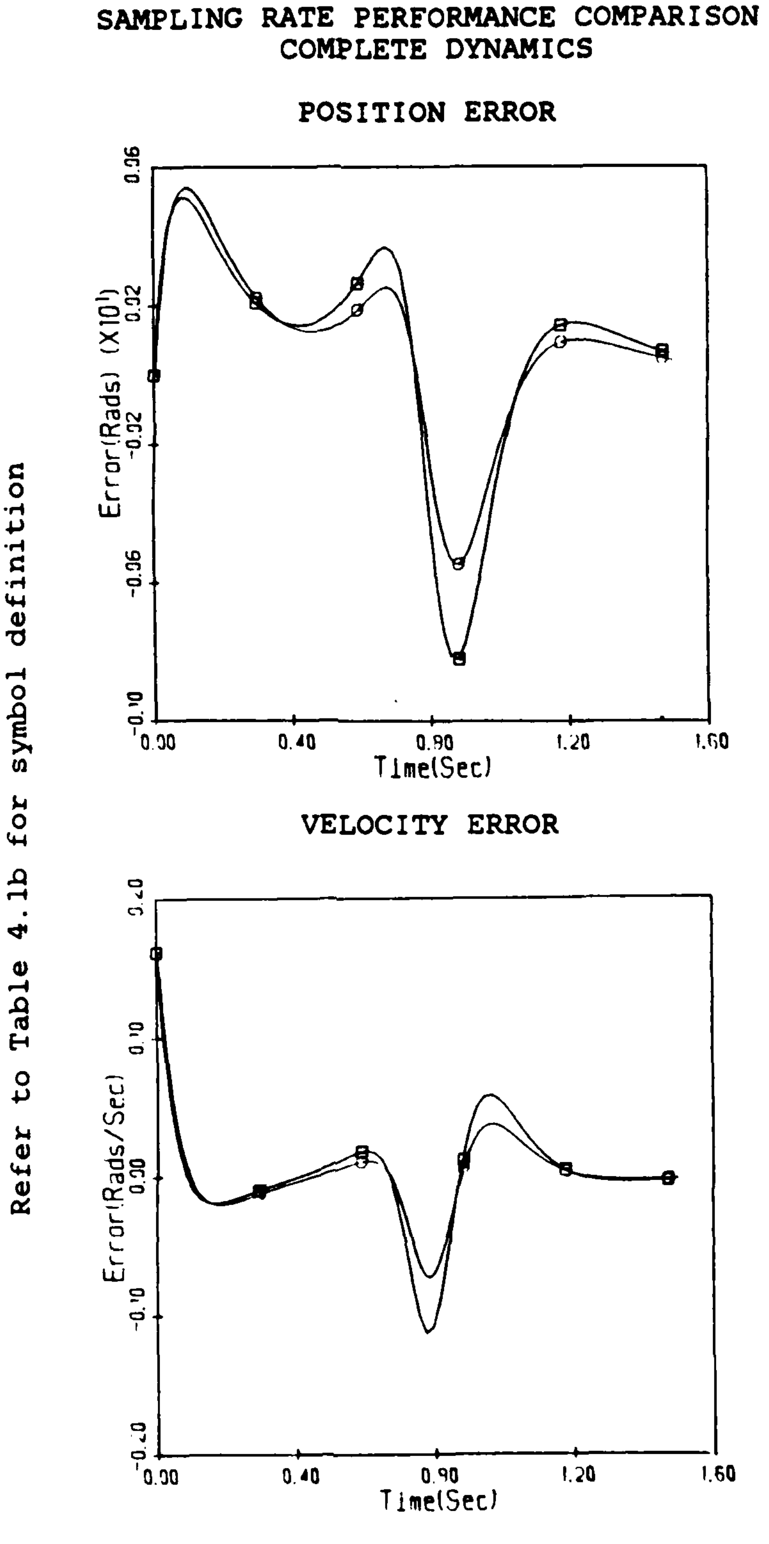
Figure 4.2e Joint 5 Fast IC1 Sample Rate Comparison

AD-A172 919 ROBOTIC MANIPULATOR CONTROL PERFORMANCE EVALUATION(U) 2/3
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH
M B LEAHY AUG 86 AFIT/CI/NR-86-173D

UNCLASSIFIED F/G 13/9 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



SAMPLING RATE PERFORMANCE COMPARISON
COMPLETE DYNAMICS

POSITION ERROR

VELOCITY ERROR

Refer to Table 4.1b for symbol definition

Figure 4.2f Joint 6 Fast IC1 Sample Rate Comparison

The most significant impact is on small link performance. Large link coupling dominates small link torque composition [44]. Extending the sampling period decreases the dynamics update rate increasing the error between the actual arm model and its control loop representation. The resultant large link performance degradation is responsible for the variation in small link accuracy.

4.4.3 Effects Of Actuator Inertias

Table 4.4 compares the computed-torque simulation performance with that of an identical simulation study where the arm model ignored the actuator inertias [52]. Rank order of large link performance is unaffected by modeling the actuators although the degree of improvement from modeling Coriolis and centrifugal forces for links 2 and 3 has been significantly reduced. Reductions in peak position error produced by actuator inertia modeling range from 0.0008 to 0.0075 radians depending on initial condition and link. Final position error differences are negligible. The initial ringing present in diagonal inertia without actuator controller velocity error [52] has been eliminated.

Variations in small link performance due to actuator modeling are significant. Actuator inertias dominate link 6 torque calculations eliminating the variation in controller performance [52] previously observed due to neglected dynamics. Joint 4 inertial dynamics algorithm's error profiles are unaltered but the peak errors decreased

TABLE 4.4

ALGORITHM SIMULATION POWER RANKING
VARIATIONS DUE TO MODELING ACTUATOR INERTIAS
FAST TRAJECTORY

	w/o actuators				with actuators			
DYNAMIC	Newton Euler	Full inertial	Block inertial	Diagonal inertial	Newton Euler	Full inertial	Block inertial	Diagonal inertial
IC0								
JOINT 1	7.50	4.74	5.05	0.00	5.96	5.54	5.63	0.00
JOINT 2	8.95	9.58	9.64	0.00	8.69	9.17	9.19	0.00
JOINT 3	8.40	7.66	8.23	0.00	8.05	8.09	8.54	0.00
JOINT 4	9.07	7.12	0.66	0.52	3.36	6.30	1.76	1.30
JOINT 5	7.65	3.56	4.80	0.56	1.51	1.80	2.23	1.00
JOINT 6	8.46	0.35	3.93	5.34	0.00	3.34	3.26	3.31
IC1								
JOINT 1	6.44	2.24	2.28	1.08	4.80	2.35	2.55	1.51
JOINT 2	8.61	8.42	8.37	0.00	8.79	8.97	8.94	0.00
JOINT 3	8.14	8.68	8.95	0.00	8.49	8.77	8.92	0.00
JOINT 4	8.64	5.15	1.52	1.77	1.70	4.96	2.46	2.62
JOINT 5	8.84	6.80	0.31	0.38	4.40	4.16	0.14	0.28
JOINT 6	8.72	0.36	5.01	3.19	0.00	3.34	3.27	3.34
IC2								
JOINT 1	0.46	5.84	5.55	1.04	0.03	6.19	5.86	2.87
JOINT 2	9.24	8.61	8.56	0.00	8.87	9.06	9.10	0.00
JOINT 3	8.79	2.44	3.30	1.42	7.95	4.00	4.32	0.01
JOINT 4	9.09	9.61	0.04	0.43	2.47	5.10	1.94	1.67
JOINT 5	9.22	9.00	0.85	1.70	3.53	5.46	1.72	1.59
JOINT 6	9.05	8.60	0.00	5.36	0.00	3.31	3.44	3.76

Power rankings illustrate relative performance by scaling and summation of the normalized peak and final position and velocity errors produced by different algorithms over identical trajectories. Power rankings range from zero to ten with the best performing algorithm annotated by the highest ranking. For additional information refer to table 4.2

more noticeably than when nonlinear forces were accounted for. Joint 5 position and velocity error peaks are reduced by factors of 5 and 2 respectively while the importance of nonlinear forces has been reduced akin to links 2 and 3.

Actuator inertias are the dominant component in small link dynamics. The effects of inertial coupling and nonlinear forces are now centered around a large constant value, greatly reducing their significance.

The important trends in large link performance can be simulated with or without modeling actuator inertias. Due to the small mass of the last three links the actuators inertias must not be ignored.

4.5 Dynamic Model Real-time Evaluation

Real-time evaluation of dynamic models for robot control is conducted over the identical trajectories employed in the simulation studies. Error data from five tests over the same trajectory are averaged for more precise assessment of each algorithm's capabilities. Efficient dynamic algorithms have been obtained from ([38],[44]). The algorithms were evaluated with and without actuator inertia modeling on a PUMA manipulator connected by an RAL Hierarchical Control System link [48] to R3AGE: The RAL Real-Time Robotic Algorithm Exerciser [40].

Implementation changes and inclusion of actuator inertias in the feedforward loop does not alter the previous conclusion [52] that simulation studies do not accurately predict arm performance. Diagonal dynamics in the feedforward loop produce superior tracking for links 2 and 3 independent of trajectory speed. Full inertial coupling in the feedforward loop produces superior joint 4 efficacy. For the other links no model consistently produces the best tracking accuracy. Small link error profiles are independent of initial condition.

4.5.1 Effects Of Inertial Coupling

Rapid changes in acceleration highlight the differences in the computed torque tracking ability due to inertial dynamics. Analysis of open-loop torque composition revealed the dominance of inertial forces for the first one second of the trajectory, and gravity thereafter [44]. Since all models utilize identical gravitational force representation performance variations are concentrated in the first second.

Real-time results validate the simulation prediction of insignificant coupling effects on link 6 performance. Simulation studies accurately forecast the minimal effect of small joint coupling on large joint performance.

In sharp contrast to the simulation predictions uncoupling manipulator dynamics produces significant improvements in controller performance for links 2 and 3. Large link coupling induces vibrations in those links when starting from the "ready" position.

Only when starting from initial condition 2 does inertial coupling aid link 1 and 5 tracking accuracy. The level of tracking improvement produced by neglecting coupling when motion starts from the other two initial conditions is significantly larger than the degradation experienced when starting from the other. Coupling has negligible repercussions on link 6 efficacy. Only for joint 4 does large link coupling consistently enhance performance.

Analysis of the closed-loop torques demonstrates that inertial coupling in the feedforward loop reduces the large joint control input. Diagonal dynamics produces the highest control input and minimum error. That relationship suggests a manipulator that resembles a series of uncoupled second order systems and not a highly coupled multivariable system.

4.5.2 Effects Of Coriolis And Centrifugal Forces

Modeling the complete dynamics in the feedforward loop doesn't produce the variations in controller performance predicted by the simulation study. Real-time individual link position error profiles produced by the complete and the full inertial dynamics are similar for the first four links from two of the three initial conditions. Knowledge of velocity related forces in the feedforward loop eliminates vibrations present in the large link when motion starts from the "ready" position and only the inertial coupling is modeled. However, complete model tracking accuracy is still inferior to the diagonal dynamics case.

Ramifications in overall controller efficacy from ignoring the Coriolis and centrifugal forces in the PUMA manipulator dynamic models are negligible. Therefore the computational savings inherent in neglecting Coriolis and centrifugal calculations are obtainable without appreciable performance penalty.

4.5.3 Effects Of Actuator Inertias

Variations in algorithm real-time power ranking produced by modeling actuator inertias are illustrated in table 4.5. Comparison data is from an identical real-time evaluation without actuator modeling. The effect of actuator inertias on link performance is more significant than predicted by simulation. Degradation of control algorithm performance due to the modeled large link inertial coupling has been reduced significantly. The diagonal dynamics average large link peak errors are reduced by 17-46 percent.

The most dramatic enhancement has been in small link algorithms performance. Without actuator modeling all four algorithms were unable to command the small links to track the desired trajectories. Small link final position

TABLE 4.5

ALGORITHM REAL-TIME POWER RANKING
VARIATIONS DUE TO MODELING ACTUATOR INERTIAS
FAST TRAJECTORY

	w/o Actuators				With Actuators			
DYNAMICS	Newton Euler	Full inertial	Block inertial	Diagonal inertial	Newton Euler	Full inertial	Block inertial	Diagonal inertial
IC0								
JOINT 1	1.19	0.33	0.27	4.70	0.82	0.40	1.45	2.65
JOINT 2	0.06	0.17	0.34	7.98	0.37	1.62	0.42	3.44
JOINT 3	0.30	0.06	1.20	7.38	0.68	0.06	0.83	2.75
JOINT 4	0.00	0.05	0.72	0.98	0.62	1.37	1.01	0.92
JOINT 5	0.96	0.93	1.51	1.94	0.00	0.43	1.01	1.37
JOINT 6	0.61	0.64	0.04	0.07	0.03	0.62	0.95	0.26
IC1								
JOINT 1	0.58	2.05	1.81	0.91	0.00	1.96	1.65	2.69
JOINT 2	1.80	1.71	0.37	5.48	0.94	0.16	0.07	4.15
JOINT 3	2.05	0.13	0.90	6.69	0.13	0.54	0.68	2.67
JOINT 4	0.00	0.04	0.09	0.06	0.43	0.92	0.57	0.57
JOINT 5	1.94	1.99	1.47	1.54	0.00	0.42	1.69	1.88
JOINT 6	0.91	0.95	0.08	0.00	0.02	0.77	0.28	0.11
IC2								
JOINT 1	2.02	3.68	1.81	1.36	0.44	1.66	1.77	1.17
JOINT 2	1.09	1.11	0.64	4.07	0.76	0.25	0.31	2.97
JOINT 3	0.36	0.60	1.15	5.49	0.00	1.92	1.93	2.44
JOINT 4	0.75	0.94	0.78	0.94	0.57	0.96	0.20	0.07
JOINT 5	1.08	1.41	0.00	0.12	1.34	1.30	0.56	1.01
JOINT 6	0.16	0.23	0.09	0.07	0.08	0.25	0.44	0.10

Power rankings illustrate relative performance by scaling and summation of the normalized peak and final position and velocity errors produced by different algorithms over identical trajectories. Power rankings range from zero to ten with the best performing algorithm annotated by the highest ranking. For additional information refer to table 4.2

PERFORMANCE COMPARISON
WITH ACTUATORS

Refer to Table 4.1b for symbol definition

SIMULATION

REAL-TIME

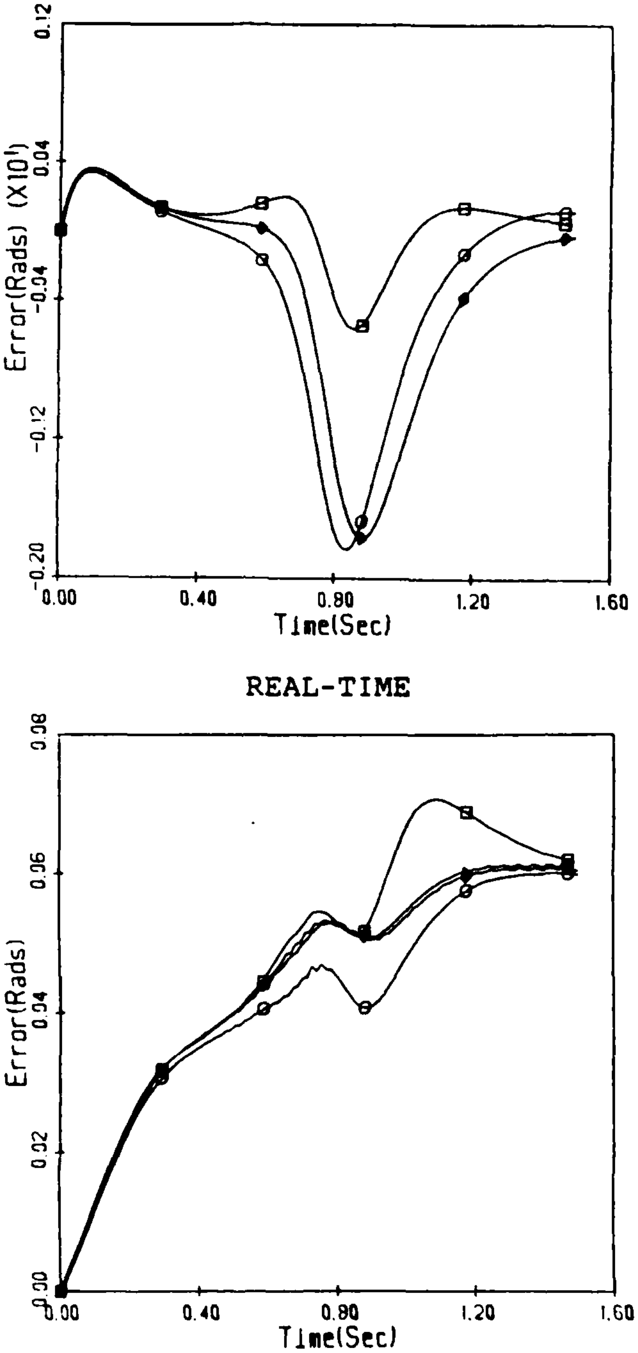
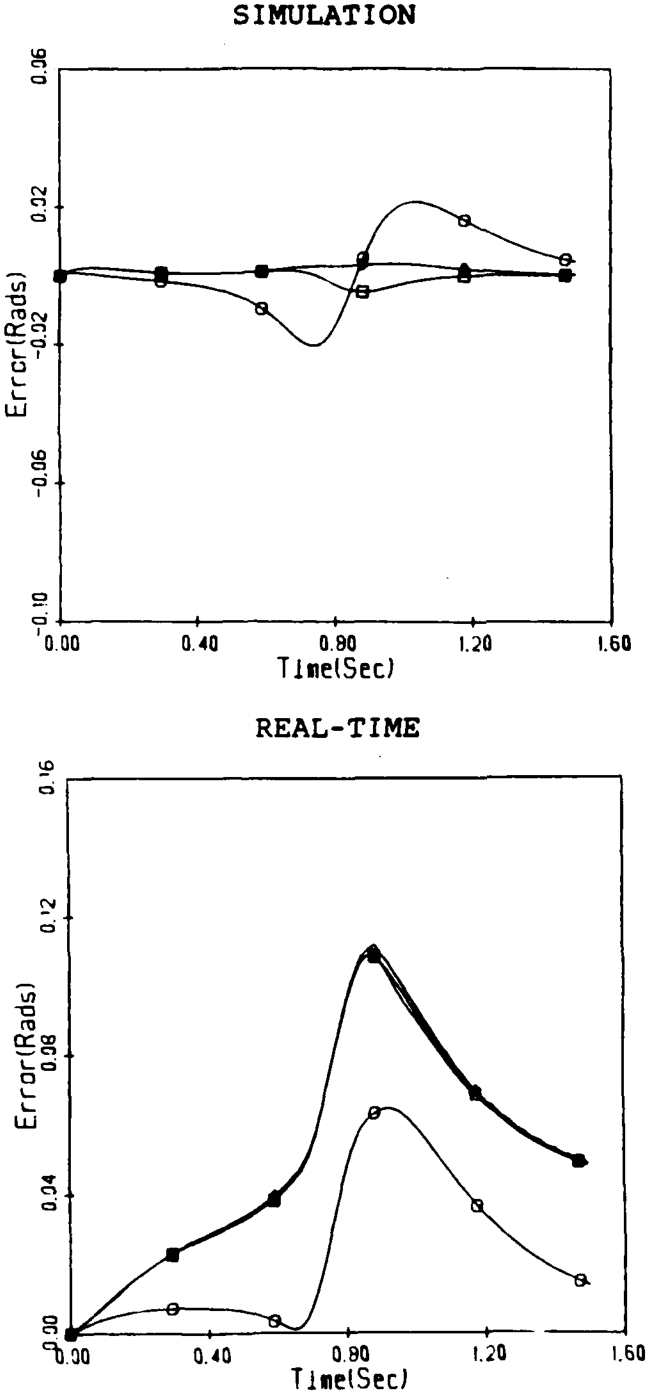


Figure 4.3a Joint 1 Fast IC1 Position Error

PERFORMANCE COMPARISON
WITH ACTUATORS

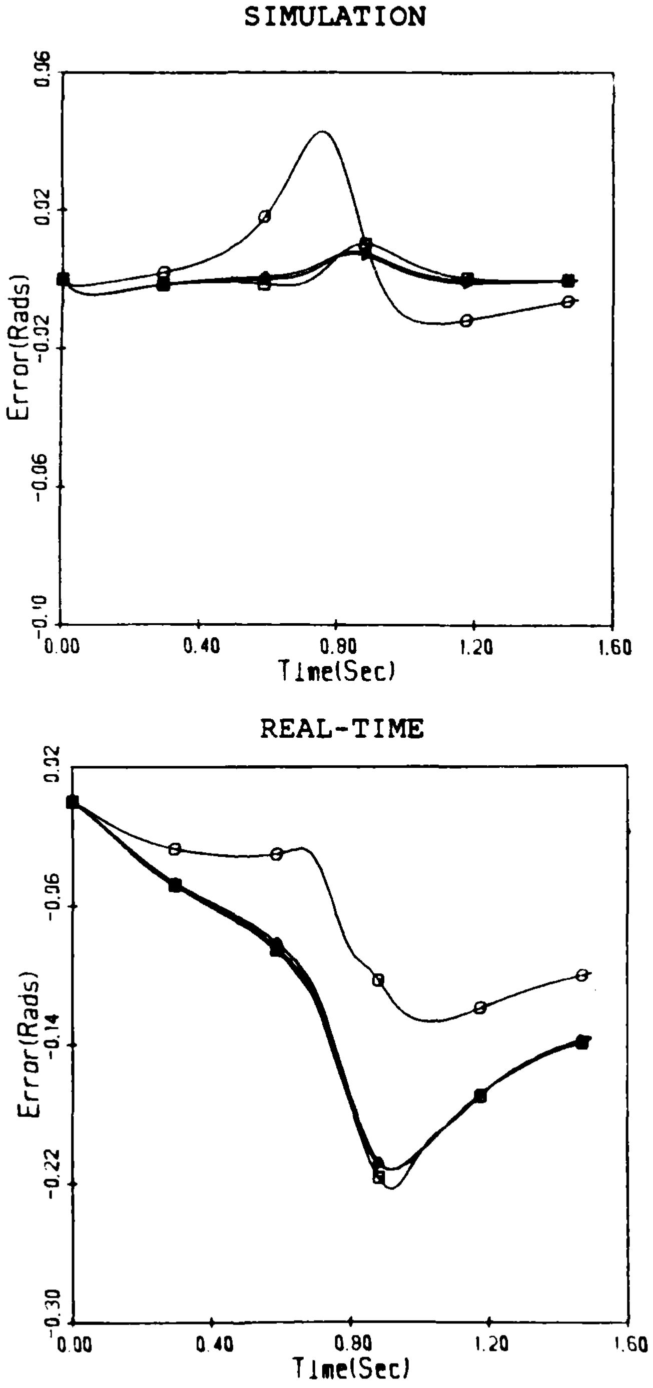

SIMULATION

REAL-TIME

Refer to Table 4.1b for symbol definition

Figure 4.3b Joint 2 Fast IC1 Position Error

PERFORMANCE COMPARISON
WITH ACTUATORS

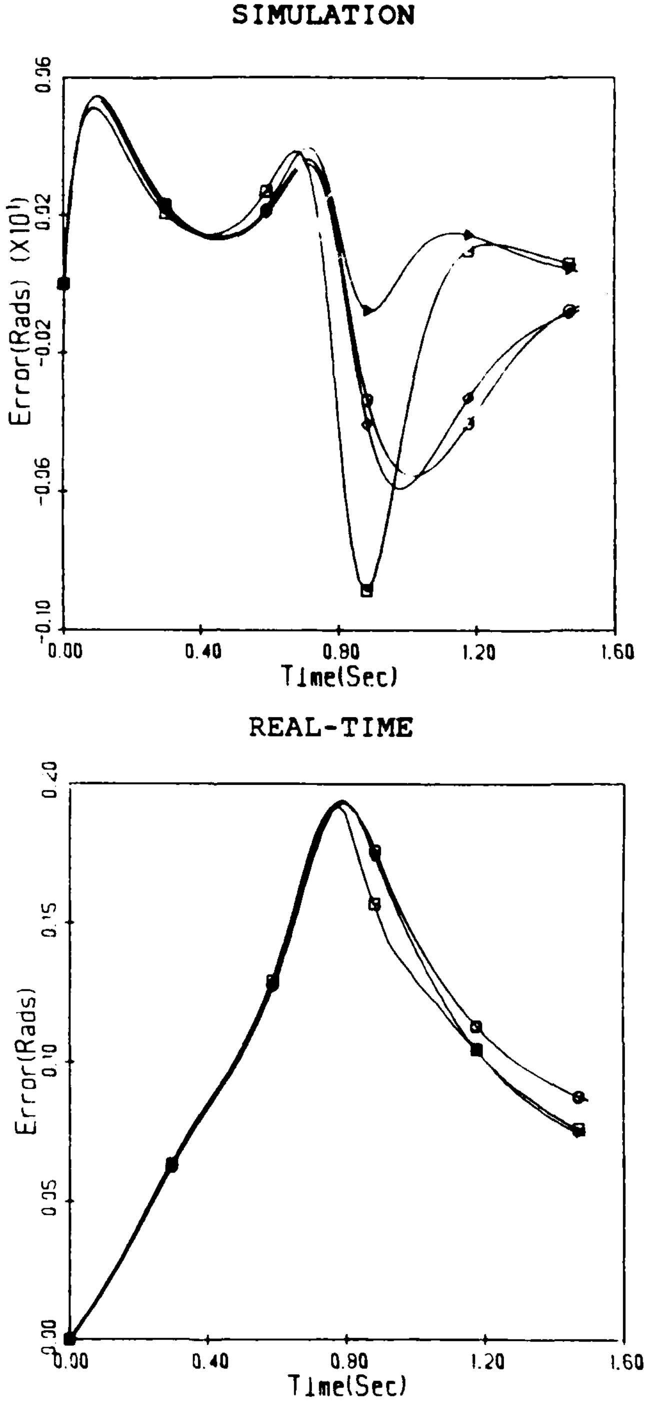

SIMULATION

REAL-TIME

Refer to Table 4.1b for symbol definition

Figure 4.3c Joint 3 Fast IC1 Position Error

PERFORMANCE COMPARISON
WITH ACTUATORS



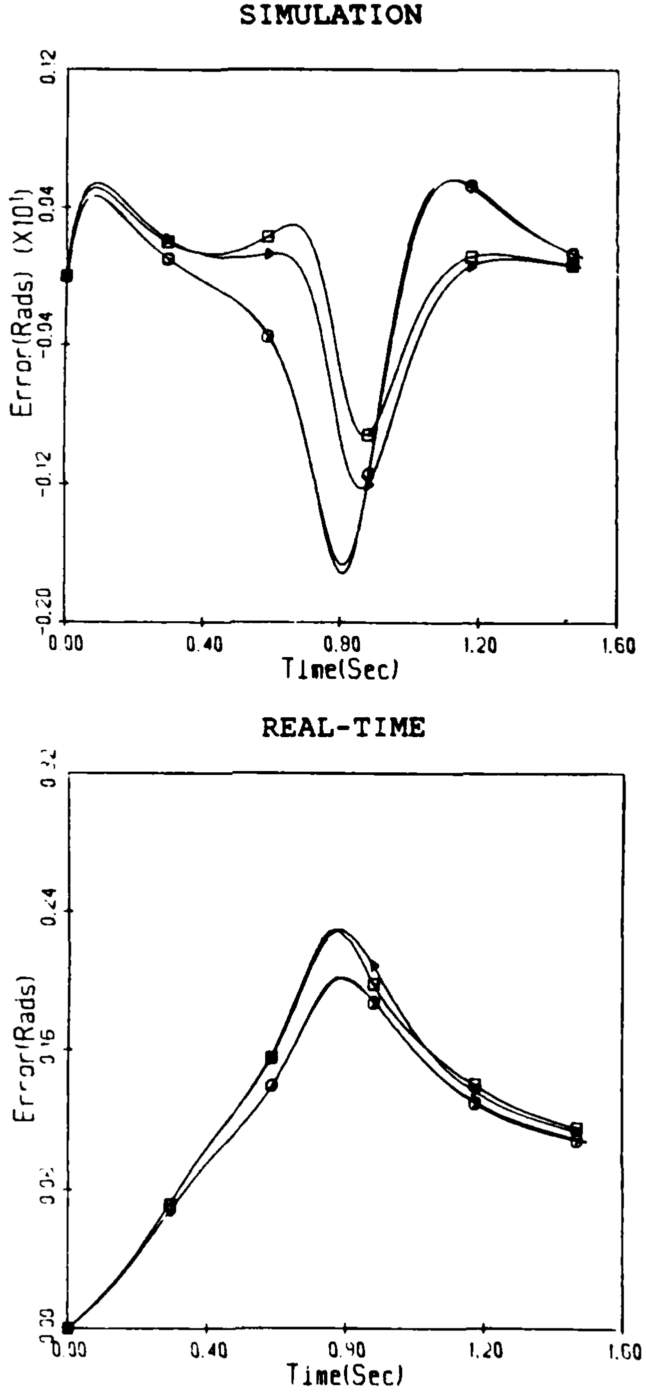
SIMULATION

REAL-TIME

Refer to Table 4.1b for symbol definition

Figure 4.3d Joint 4 Fast IC1 Position Error

PERFORMANCE COMPARISON
WITH ACTUATORS

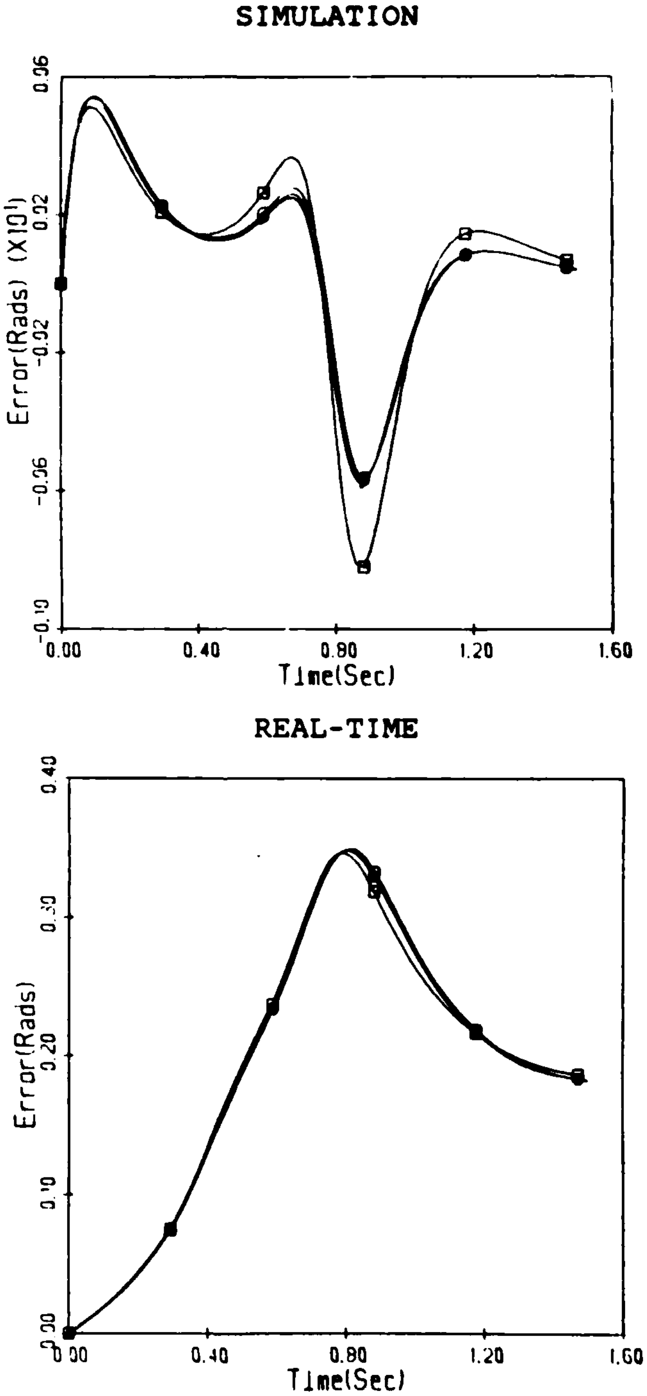

SIMULATION

REAL-TIME

Refer to Table 4.1b for symbol definition

Figure 4.3e Joint 5 Fast IC1 Position Error

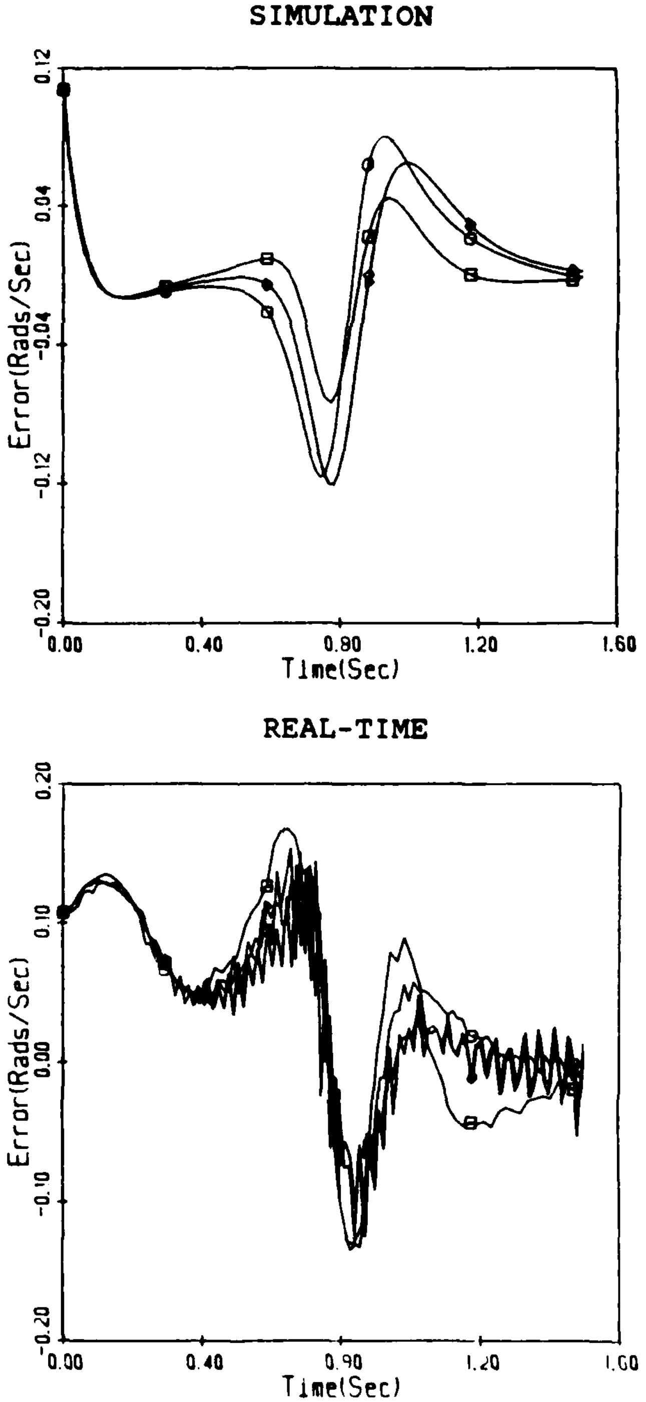
PERFORMANCE COMPARISON
WITH ACTUATORS



Refer to Table 4.1b for symbol definition

Figure 4.3f Joint 6 Fast IC1 Position Error

PERFORMANCE COMPARISON
WITH ACTUATORS

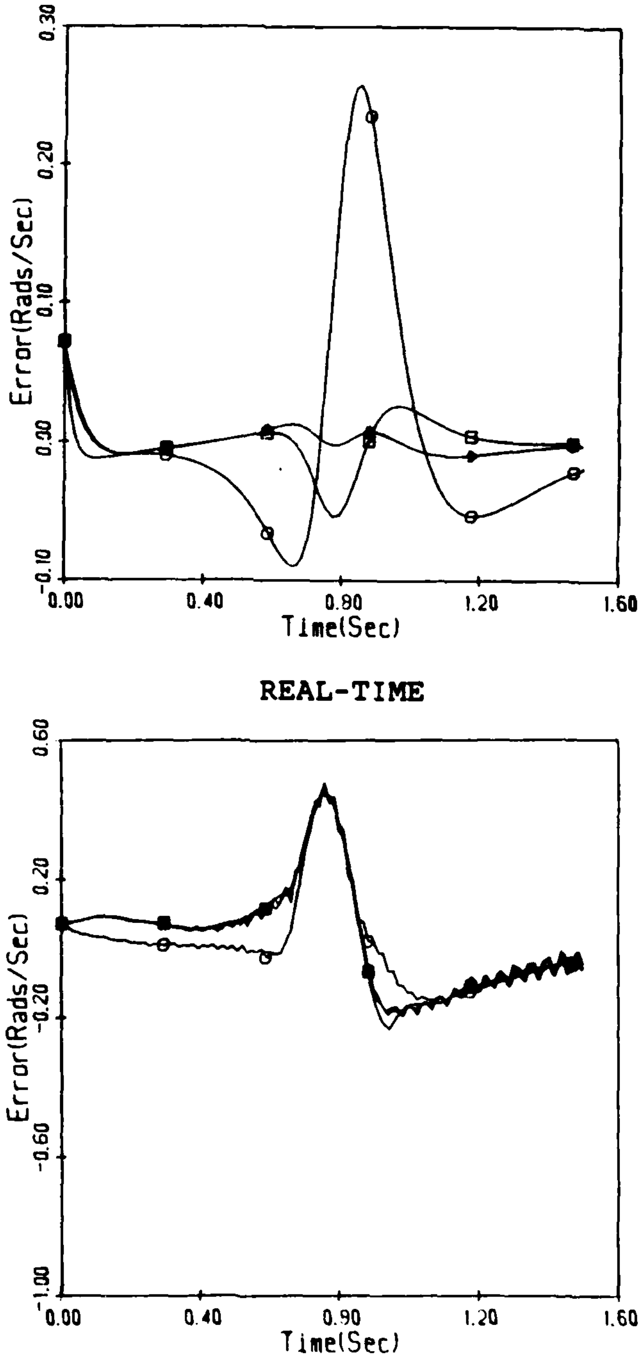

SIMULATION

REAL-TIME

Refer to Table 4.1b for symbol definition

Figure 4.4a Joint 1 Fast IC1 Velocity Error

PERFORMANCE COMPARISON
WITH ACTUATORS

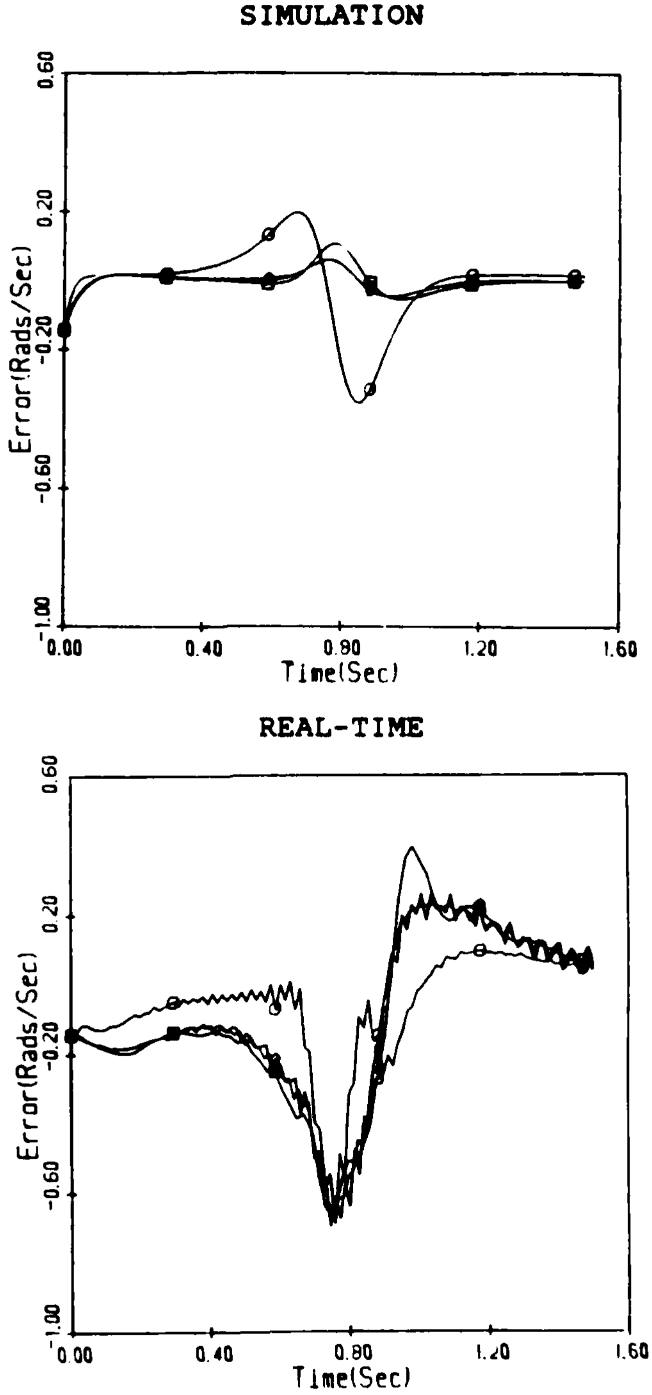

SIMULATION

REAL-TIME

Refer to Table 4.1b for symbol definition

Figure 4.4b Joint 2 Fast IC1 Velocity Error

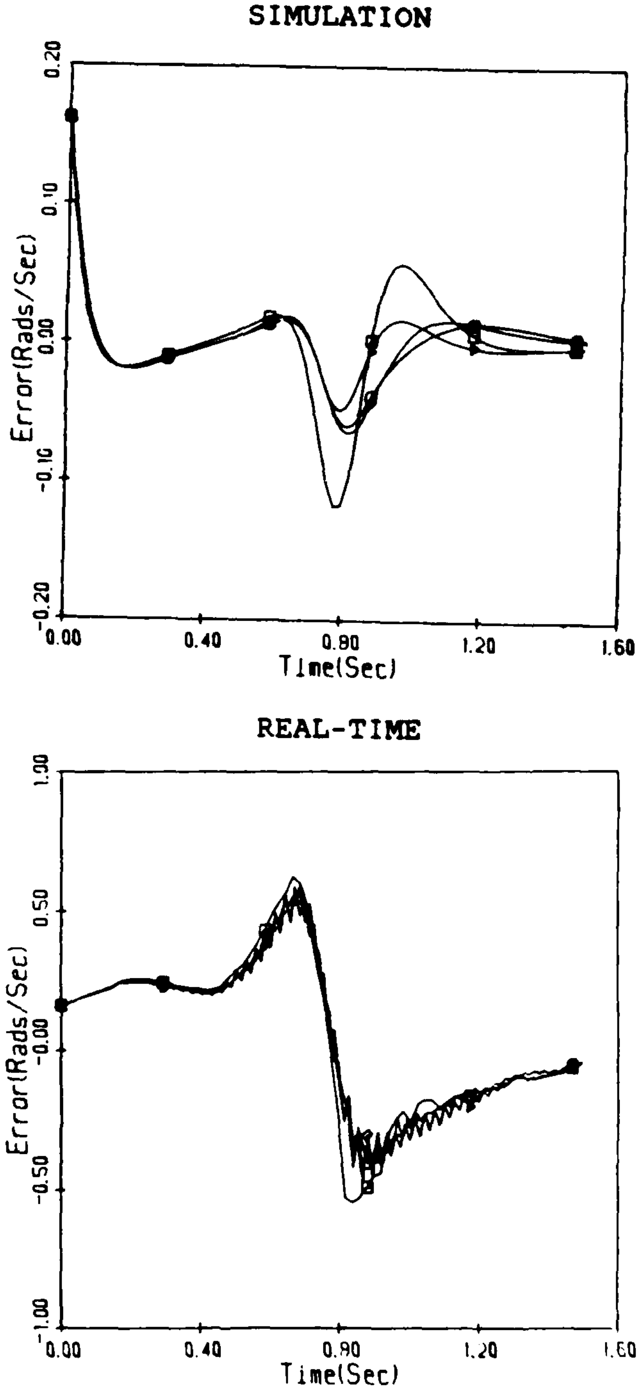
PERFORMANCE COMPARISON
WITH ACTUATORS



Refer to Table 4.1b for symbol definition

Figure 4.4c Joint 3 Fast IC1 Velocity Error

PERFORMANCE COMPARISON
WITH ACTUATORS

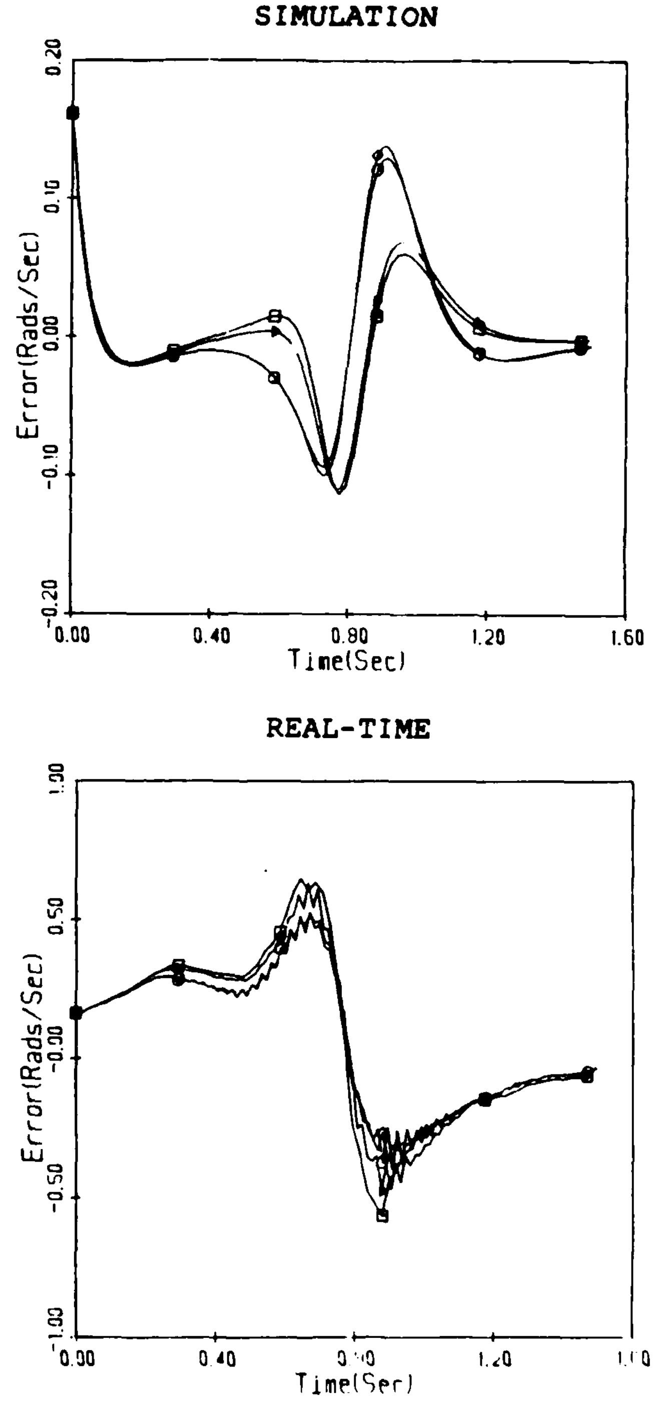

SIMULATION

REAL-TIME

Refer to Table 4.1b for symbol definition

Figure 4.4d Joint 4 Fast IC1 Velocity Error

PERFORMANCE COMPARISON
WITH ACTUATORS

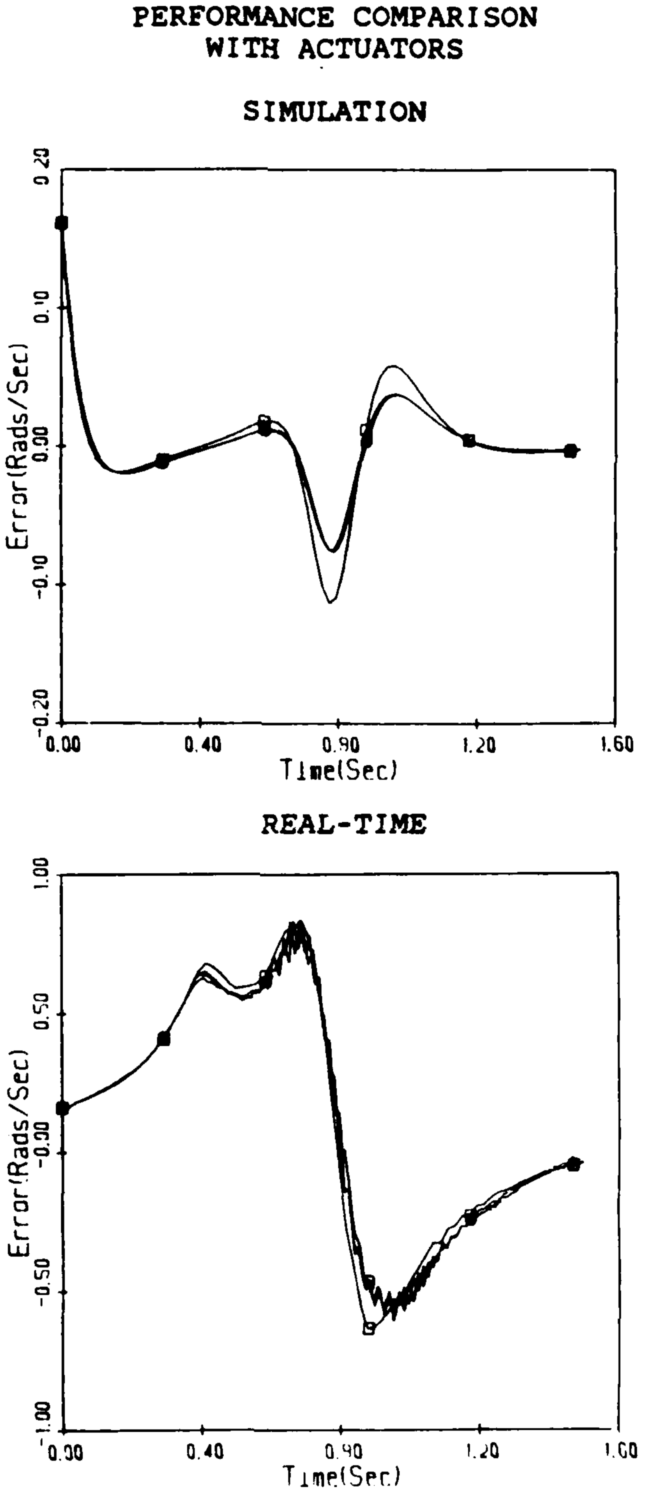


SIMULATION

REAL-TIME

Refer to Table 4.1b for symbol definition

Figure 4.4e Joint 5 Fast IC1 Velocity Error



PERFORMANCE COMPARISON
WITH ACTUATORS

SIMULATION

REAL-TIME

Refer to Table 4.1b for symbol definition

Figure 4.4f Joint 6 Fast IC1 Velocity Error

errors were over 1.20 radians. By modeling actuator inertias link 4, 5, and 6 average final position error are reduced to 0.8, 0.1 and 0.3 radians respectively.

4.6 Discussion

The heuristic global linearization scheme of the computed-torque technique produces a control law analogous to the mathematically based exact linearization [95], nonlinear feedback [20], and optimal control methods [65]. Therefore the results of this research are applicable to a whole class of dynamically based linearization techniques.

Large link simulation results are analogous to a similar study performed on a TART manipulator [98]. The simulated performance of the computed-torque technique degrades as the accuracy of the dynamic model is decreased. However, as table 4.6 illustrates, simulation results do not accurately identify the effects of dynamic models on real-time computed-torque technique performance.

The ability of the computed-torque technique to asymptotically drive trajectory tracking errors to zero is based on the assumption that the manipulator dynamics can be globally linearized. Only in the presence of accurate modeling is equation 4.4 valid. Modeling inaccuracies are reflected in the feedforward loop' s inability to completely cancel the manipulator nonlinear dynamics introducing perturbations into the feedback loop. If the feedback loop is not robust enough to reject those disturbances the

TABLE 4.6

ALGORITHM POWER RANKING
SIMULATION AND REAL-TIME COMPARISON
FAST TRAJECTORY

	Simulation				Real-Time			
DYNAMICS	Newton Euler	Full inertial	Block inertial	Diagonal inertial	Newton Euler	Full inertial	Block inertial	Diagonal inertial
IC0								
JOINT 1	5.96	5.54	5.63	0.00	0.82	0.40	1.45	2.65
JOINT 2	8.69	9.17	9.19	0.00	0.37	1.62	0.42	3.44
JOINT 3	8.05	8.09	8.54	0.00	0.68	0.06	0.83	2.75
JOINT 4	3.36	6.30	1.76	1.30	0.62	1.37	1.01	0.92
JOINT 5	1.51	1.80	2.23	1.00	0.00	0.43	1.01	1.37
JOINT 6	0.00	3.34	3.26	3.31	0.03	0.62	0.95	0.26
IC1								
JOINT 1	4.80	2.35	2.55	1.51	0.00	1.96	1.65	2.69
JOINT 2	8.79	8.97	8.94	0.00	0.94	0.16	0.07	4.15
JOINT 3	8.49	8.77	8.92	0.00	0.13	0.54	0.68	2.67
JOINT 4	1.70	4.96	2.46	2.62	0.43	0.92	0.57	0.57
JOINT 5	4.40	4.16	0.14	0.28	0.00	0.42	1.69	1.88
JOINT 6	0.00	3.34	3.27	3.34	0.02	0.77	0.28	0.11
IC2								
JOINT 1	0.03	6.19	5.86	2.87	0.44	1.66	1.77	1.17
JOINT 2	8.87	9.06	9.10	0.00	0.76	0.25	0.31	2.97
JOINT 3	7.95	4.00	4.32	0.01	0.00	1.92	1.93	2.44
JOINT 4	2.47	5.10	1.94	1.67	0.57	0.96	0.20	0.07
JOINT 5	3.53	5.46	1.72	1.59	1.34	1.90	0.56	1.01
JOINT 6	0.00	3.31	3.44	3.76	0.08	0.25	0.44	0.10

Power rankings illustrate relative performance by scaling and summation of the normalized peak and final position and velocity errors produced by different algorithms over identical trajectories. Power rankings range from zero to ten with the best performing algorithm annotated by the highest ranking. For additional information refer to table 4.2

tracking error will not be zero. For these evaluations the feedback loop was a constant and the feedforward loop dynamics were varied. Therefore the feedforward formulation that produces the minimum control errors best models the manipulator.

In the simulation study the best overall trajectory tracking accuracy was achieved by employing the full inertia model in the feedforward loop. For real-time applications the best overall trajectory tracking performance was achieved by modeling the PUMA as follows:

$$T_i = \sum_{j=i}^{6} D_{ij}\ddot{q}_j + I_{ai}\ddot{q}_i + D_i$$

Where:

T_i = Torque acting at joint i

q_i = ith joint position

$\ddot{q}_i$ = Acceleration of ith joint

D_{ii} = Effective inertia at joint i

D_{ij} = Coupling inertia between joint i and j when $i \neq j$, zero for all $i \neq 4$

D_i = Gravity loading at joint i

I_{ai} = Actuator inertia

Therefore the actual PUMA arm is not a highly coupled nonlinear system. Control algorithm comparison studies employing the complete Lagrange or Newton-Euler models to simulate the PUMA manipulator produce invalid conclusions.

An evaluation of computed-torque performance on a direct drive manipulator conducted at CMU [34] reveals that overmodeling of system dynamics is not a PUMA specific phenomenon. Even with the lack of gearing and friction, complete knowledge of manipulator dynamics produced tracking accuracy inferior to the performance of an identical algorithm utilizing a reduced form of dynamics in it's feedforward loop.

Even with the best dynamical model, PUMA computed-torque performance was unacceptable for implementation as a gross motion controller. Sweet and Good suggest that drive system interactions dominated the actual dynamics of a manipulator with harmonic drives [93]. Results presented here indicate that unmodeled forces such as friction or drive system interactions may dominate actual arm dynamics. Their influence on robot control was too significant to be effectively compensated for by the feedback loop employed in this study.

Although the performance level was unacceptable these results do illustrate the robust nature of the computed-torque technique to parameter uncertainties. Modeling errors produced by overcompensating for the impact of inertial coupling produced higher tracking errors, not instability. Implementation errors resulting in a 180 degree difference between the modeled and actual locations of joints 1 and 3 were unable to produce unstable behavior [52].

Therefore the error optimizing technique of Bejczy, Tarn et.al. ([10],[95]) is not required to assure stability of an exactly linearized system.

The robust nature of the computed-torque response is partly due to the high degree of mechanical damping inherent in a PUMA manipulator. The high gear ratios and actuator inertia dominance enhance the stability of the PUMA manipulator. Direct drive manipulators do not duplicate those traits and therefore are more susceptible to parameter uncertainties.

4.7 Summary

A significant contribution to the manipulator control database has been accomplished. For the first time dynamic models for robotic control have been evaluated by simulated and real-time implementation of four forms of dynamics in a computed-torque algorithm. The results from the evaluation of dynamics for robot control are summarized in table 4.7.

The evaluation of dynamics for robot control by simulation of a dynamics based control law has revealed that:

1. algorithm performance is directly dependent on large link coupling information,
2. large link performance is independent of small link inertial coupling information,

3. actuator inertias are the dominant term in small link modeling,
4. small link performance is minimally dependent on small link inertial coupling information, and
5. increased sampling times necessary for Newton-Euler implementation largely offset advantages of Coriolis and centrifugal modeling.

The evaluation of dynamics for robot control by real-time implementation of a dynamics based control law has revealed that:

1. the effects of Coriolis and centrifugal forces are negligible,
2. unmodeled forces cancel the benefits of inertial coupling displayed in the simulation study for all links except the fourth,
3. inclusion of reflected actuator inertias in the feedforward loop significantly enhances tracking accuracy especially for the small links,
4. gravity forces are significant and should be modeled in the feedforward loop, and
5. diagonal inertial terms are significant and should be modeled in the feedforward loop.

Real-time results contradict simulation conclusions. The simulation conclusions are valid for a direct drive version of the PUMA but not for the highly geared friction dependent device currently available. The real-time results may be extended to harmonic drives and the small links of a direct drive manipulator.

TABLE 4.7

ALGORITHM POWER RANKING
SIMULATION AND REAL-TIME OVERALL COMPARISON

	SIMULATION				REAL-TIME			
DYNAMICS	NEWTON EULER	FULL INERTIAL	BLOCK INERTIAL	DIAGONAL INERTIAL	NEWTON EULER	FULL INERTIAL	BLOCK INERTIAL	DIAGONAL INERTIAL
SLOW	4.78	5.72	4.71	1.19	1.61	1.27	1.27	2.65
FAST	4.36	5.55	4.63	1.29	0.40	0.90	0.88	1.73
OVERALL	4.57	5.64	4.67	1.24	0.50	1.08	1.08	2.19

Slow and Fast values represent power ranking data averaged over all initial conditions and joints. Overall values average Slow and Fast data.

Even with the best dynamical model, computed-torque performance was unacceptable for utilization as a real-time gross motion controller. The importance of unmodeled forces clearly illustrates the requirement for better feedforward modeling and/or feedback compensation techniques if dynamics based control methods are to be successfully employed as gross motion controllers. The implementation feasibility and performance improvement potential of unmodeled force compensation techniques is the subject of the next chapter.

CHAPTER 5

COMPENSATION OF UNMODELED MANIPULATOR DYNAMICS

5.1 Introduction

The evaluation of dynamics for robot control in chapter four illuminated the pivotal role of forces unmodeled by Lagrange-Euler dynamics in controller accuracy. Techniques for eradication of the effects of those forces are necessary if dynamics based control methods are to be successfully applied to robotic manipulators. Therefore, knowledge of the implementation feasibility and performance improvement potential of unmodeled force compensation techniques would be an invaluable aid in the design of modern robotic control laws.

The computed-torque technique is the most basic representation of the dynamically dependent control philosophy. The heuristic global linearization scheme of the computed-torque technique produces a control law analogous to the mathematically based exact linearization [95], nonlinear feedback [20], and optimal control methods [65]. Knowledge about the implementation feasibility and performance potential of unmodeled force compensation techniques can be obtained from evaluation of the performance ramifications produced by incorporation of those techniques into the computed-torque control law.

In this chapter an important contribution to the manipulator control database is conducted by an evaluation of feedforward and feedback techniques for compensation of PUMA manipulator unmodeled forces. The impact of improved feedforward modeling of the manipulator on computed-torque accuracy is evaluated along with feedforward friction compensation. Two new control strategies are applied in the feedback loop to determine if increased feedback gain can eliminate the disturbances resulting from the forces unmodeled by the feedforward loop.

Utilization of more accurate manipulator inertia parameters did not significantly improve controller effectiveness. Friction compensation by a nonlinear switching function produces severe large joint vibration while increasing small link accuracy. A higher bandwidth feedback loop improves the accuracy of all joints. The most significant performance enhancement is a fifty percent reduction in small link maximum peak and final position errors. A computed-torque/PID control technique confines all joint's maximum final position errors to under one degree while producing a maximum peak position error of under five degrees.

5.2 Method Of Approach

The computed-torque technique ([55],[78],[84]) employs both feedforward and feedback elements to control a robot arm and is a special case of the optimal control law

[65]. In chapter four that control technique formed the basis for the study of the effects of dynamics on real-time robotic control. In that study even the best dynamical formulation was unable to reduce the tracking errors sufficiently for realistic gross motion implementation. That research will be extended to determine if the computed-torque technique can be modified with non-sensor based techniques to produce tracking accuracy within acceptable limits. The limits were selected as three degrees maximum peak and one degree maximum final position error. That degree of error will allow gross motion control to position the manipulator end-effector into a sphere around the desired final position where sensor driven techniques can be applied for fine motion control. By evaluating PUMA manipulator performance variations the performance improvement potential of computed-torque feedforward and feedback loop compensation techniques on gross motion joint control are exposed.

To obtain comprehensive information about the performance improvement potential of the computed-torque technique the compensated algorithms have been evaluated over the six different operational environments employed in chapter four. The six test configurations can be broken down into two blocks:

1. slow trajectory unloaded, and

2. fast trajectory unloaded.

Each block consists of three separate trajectories with identical velocity and acceleration profiles but different initial positions. The three sets of initial conditions (IC0,IC1,IC2) are displayed in table 5.1 along with a data key. The fast trajectories shown in figure 4.1 are derived from a performance characterization study of the PUMA arm [43]. The peak velocity of each joint is achieved while avoiding real-time acceleration and torque saturation effects [43]. The slow trajectory has identical final positions but reduced velocities and accelerations due to a 25 percent elongation of the trajectory time. Joint 3 trajectories are reversed so that links 2 and 3 rotate in the same direction. For initial condition 2 (IC2) the joint trajectories are reversed to evaluate motion against gravity.

Real-time control algorithm evaluation is accomplished through utilization of the RAL Real-Time Robotic Algorithm Exerciser, R3AGE ([40],[48]). A 14ms sampling rate is selected for all algorithms. Error data from five tests over the same trajectory are averaged for more precise assessment of each compensation technique's capabilities. To quantitatively compare the effects of feedforward and feedback compensation on robot control algorithm performance the power ranking formula shown in table 4.2 is again employed.

TABLE 5.1a CHAPTER 5 DATA KEY

TITLE = XCTISMT

X - Test type.

N - Real-time RAL inertial parameters

T - Real-time TARN inertial parameters

CT - Control algorithm identifer.

12 - Diagonal inertia dynamics

50 - Diagonal inertia dynamics with friction

52 - Diagonal inertia dynamics with doubled pole PD

54 - Diagonal inerita dynamics with PID

I - Initial condition specifier

0 - IC0 (0,-90,90,0,1,0,)

1 - IC1 (0,-135,135,0,1,0)

2 - IC2 (90,0,0,90,90,90)

S - Trajectory speed specifier

0 - Slow speed

1 - Fast speed

M - External load specifier

0 - unloaded

1 - fully loaded(2.3kg)

T - Sampling time specifier

1 - 7ms

2 - 14ms

3 - 21ms

TABLE 5.1b CHAPTER 5 SYMBOL KEY

⊡	N121102
⊙	T121102
×	T501102
⊠	T521102
★	T541102

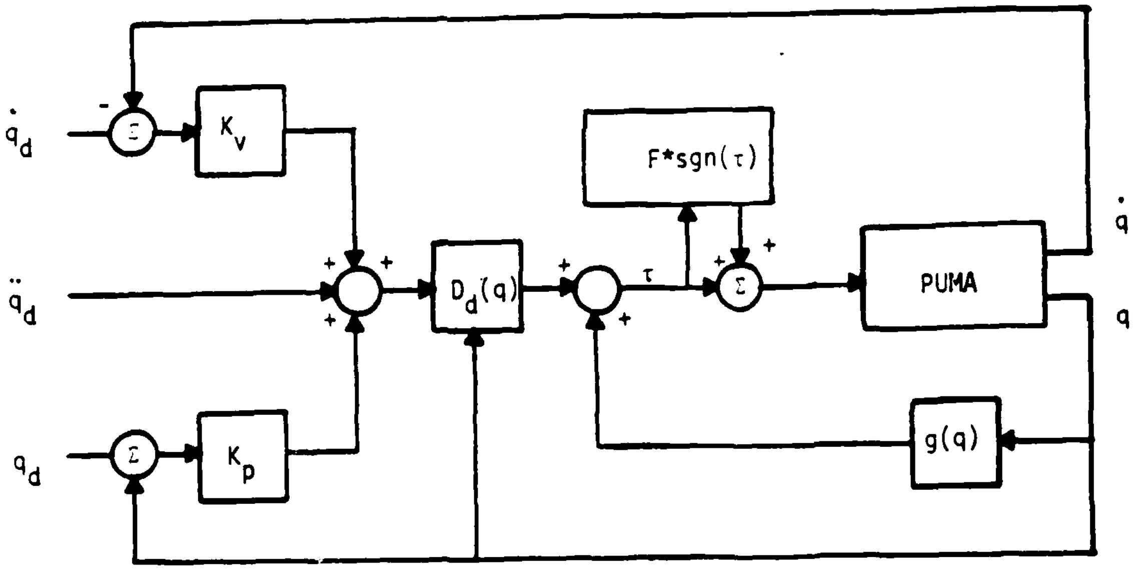


Figure 5.1a Diagonal Inertia, Gravity and Friction Feedforward Dynamics Computed-Torque Block Diagram

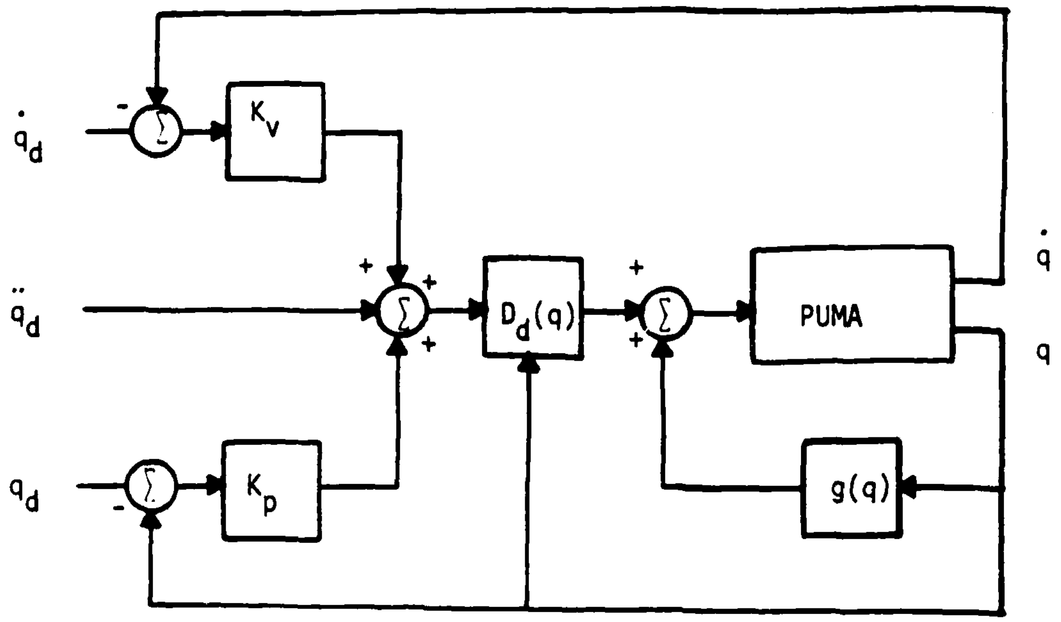


Figure 5.1b Diagonal Inertia with Gravity Feedforward Dynamics *Doubled Pole Feedback Loop Computed-Torque Block* Diagram

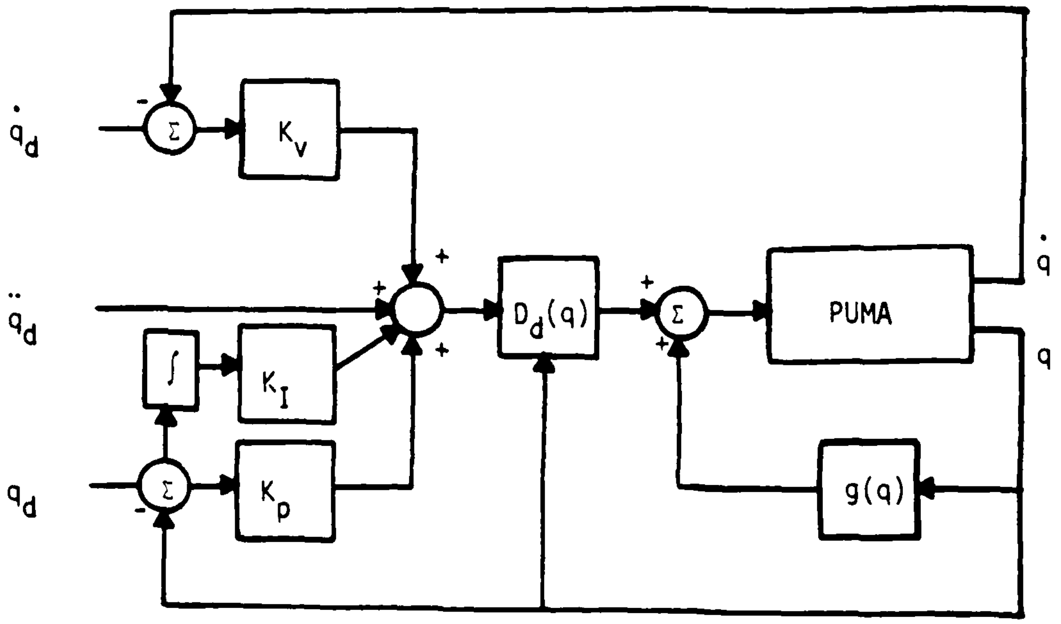


Figure 5.1c Diagonal Inertia with Gravity Feedforward Dynamics PID Feedback Loop Computed-Torque Block Diagram

The essential conclusions about the effects of feedforward and feedback compensation techniques on robot control algorithm performance are extracted from analysis of controller effectiveness in tracking the fast trajectory starting from different initial conditions. By employing the various starting points, the masking of important trends by gravity and other position dependent forces is avoided.

5.3 Computed-torque Compensation Techniques

Block diagrams of the compensated computed-torque techniques evaluated in this study are illustrated in figure 5.1. Table 5.2 displays the continuous transfer function and pole locations of the feedback loops.

The two forms of feedforward compensation examined are improved manipulator inertial parameters measurements and modeling of static friction. In these examinations the feedback loop is identical to the one utilized in chapter four. The velocity and position gain matrices are equal for each joint and have values of 20 and 100 respectively, placing the system poles at -10. The more accurate inertial parameters are from a recent study by Tarn et. al. [94]. Those parameters are incorporated into the general Lagrange-Euler dynamics algorithm [38]. The effects of the new parameters on the open-loop torque generation capability of the complete, full and diagonal inertia matrix with gravity but without Coriolis and centrifugal terms, Lagrange-Euler manipulator dynamics are determined. Those

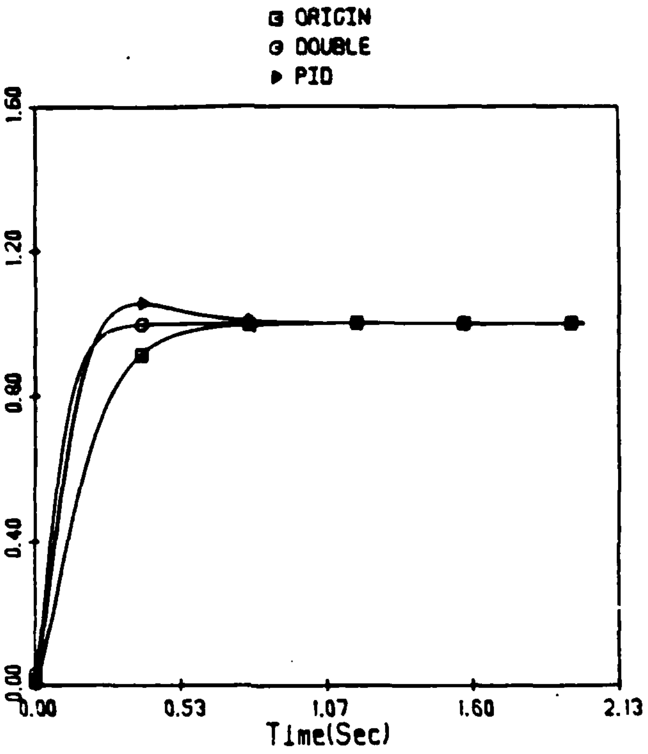


Figure 5.2 Feedback Loop Step Response Comparison

TABLE 5.2

COMPUTED-TORQUE FEEDBACK LOOP
TRANSFER FUNCTIONS

	H(S)	S-PLANE POLES
Original Eq. 4.1 & 5.1	$\frac{1}{S^2+20S+100}$	-10, -10
Doubled Pole PD Eq. 5.2	$\frac{1}{S^2+40S+400}$	-20, -20
PID Eq. 5.3	$\frac{1}{S^3+30S^2+300S+1000}$	-10, -10, -10

effects are contrasted with the ramifications from utilization of the parameters employed in chapter four's study. All dynamical formulations include actuator inertias. The constant calculator segment of the efficient dynamics algorithms [38-9] is modified so that the repercussions of parameter alterations on the evaluation of dynamics for robot control can be studied in real-time.

Static friction in the PUMA gear trains produces a torque deadband for each joint. Limits on that deadband have been determined in ([13],[43]). Figure 4.3 illustrates high initial position errors that could be the product of a lack of accurate static friction compensation. The feasibility of reducing that high initial position error by an additive friction function in the feedforward loop is evaluated.

Zhang and Paul were the first researchers to propose static friction compensation by a nonlinear switching function [111]. Addition of a nonlinear switching function in the feedforward loop changes the computed-torque control law to:

$$\tau(t) = D_d(q)[\ddot{q}_d(t)+K_v{}^T(\dot{q}_d(t)-\dot{q}(t))+K_p{}^T(q_d(t)-q(t))] \\ +g(q)+F^T \operatorname{sgn}(\tau(t)) \qquad (5.1)$$

Where:

$\tau(t)$ = Vector of joint torques

q_d, $\dot{q}_d$, $\ddot{q}_d$ = Vectors of desired position, velocity, and acceleration in generalized joint coordinates

q, $\dot{q}$, $\ddot{q}$ = Vectors of measured position, velocity and acceleration in generalized joint coordinates

$D_d(q)$ = Vector of diagonal and actuator inertias

K_v = Vector of derivative feedback gains

K_p = Vector of position feedback gains

$g(q)$ = Vector of gravity forces

F = Vector of friction compensation torques

$\operatorname{sgn}(\tau(t))$ = Vector of torque signs (+1 or -1)

Equation 5.1 is illustrated in block diagram form by figure 5.1a. The switching function limits are from a performance characterization of the PUMA [43].

A more rigorous analysis of the position errors produced in chapter four reveals that the error profiles are indicative of a control system whose frequency response is inadequate to track the desired input trajectory. Two methods of improving the frequency response of the computed-torque loop are examined. In the first method the improved inertial model is coupled with a feedback loop where the original PD poles are shifted to the left. The doubled pole PD control law is:

$$\tau(t)=D_d(q)[\ddot{q}_d(t)+K_v{}^T(\dot{q}_d(t)-\dot{q}(t))+K_p{}^T(q_d(t)-q(t))]+g(q) \qquad (5.2)$$

Where:

$\tau(t)$ = Vector of joint torques

q_d, $\dot{q}_d$, $\ddot{q}_d$ = Vectors of desired position, velocity, and acceleration in generalized joint coordinates

q, $\dot{q}$, $\ddot{q}$ = Vectors of measured position, velocity and acceleration in generalized joint coordinates

$D_d(q)$ = Vector of diagonal and actuator inertias

K_v = Vector of derivative feedback gains

K_p = Vector of position feedback gains

$g(q)$ = Vector of gravity forces

A block diagram representation of equation 5.2 is displayed in figure 5.1b. The expression for the doubled pole PD computed-torque control law is identical to equation 4.1. The control laws of equations 4.1 and 5.2 differ in the selection of the feedback gains. The velocity and position gains are still equal for each joint but for the doubled pole case they have values of 40 and 400 respectively, placing the system poles at -20. The step response of the doubled pole and original feedback loops is compared in figure 5.2.

Luo and Saridis showed that a manipulator could be controlled by an optimal/PID formulation of the computed-torque control law [65-6]. Furuta et. al. applied suboptimal computed-torque/PID control to a three degree of freedom manipulator [21]. The computed-torque/PID control law for the PUMA is:

$$\tau(t) = D_d(q)[\ddot{q}_d(t)+K_v{}^T(\dot{q}_d(t)-\dot{q}(t))+K_p{}^T(q_d(t)-q(t))$$
$$+K_I{}^T\int(q_d(t)-q(t))]+g(q) \qquad (5.3)$$

Where:

$\tau(t)$ = Vector of joint torques

$q_d{}^k\ \dot{q}_d,\ \ddot{q}_d$ = Vectors of desired position, velocity, and acceleration in generalized joint coordinates

$q,\ \dot{q},\ \ddot{q}$ = Vectors of measured position, velocity and acceleration in generalized joint coordinates

$D_d(q)$ = Vector of diagonal and actuator inertias

K_v = Vector of derivative feedback gains

K_p = Vector of position feedback gains

K_I = Vector of integral feedback gains

$g(q)$ = Vector of gravity forces

A block diagram representation of equation 5.3 is displayed in figure 5.1c. Optimal gain selection is not investigated during this initial six degree of freedom manipulator evaluation. The position, derivative and integral gains are equal for each joint and have values of 300, 30 and 1000 respectively, placing a triple pole at -10. PID step response is compared to the original and doubled pole performance in figure 5.2.

5.4 Improved Inertial Modeling Evaluation

After the completion of the evaluation of dynamics for robot control presented in chapter four, researchers at Washington University lead by Tarn completely disassembled and modeled a PUMA manipulator. The product of that effort was a more accurate set of inertial parameters ([94],[96]). Knowledge of the repercussions from more accurate Lagrange-Euler dynamics on robot control would provide a valuable contribution to the manipulator control database.

Real-time evaluations revealed that more accurate representation of PUMA inertial parameters does not alter chapter four's conclusion that uncoupled dynamics produce the best controller performance. Increased parameter accuracy in the diagonal dynamics trades off improvement in joint 3 performance for slight degradation in joint 2 accuracy. For the small links the dominance of the actuator inertias renders improved modeling of other parameters irrelevant.

The comparison of chapter four and improved parameter diagonal feedforward computed-torque fast trajectory IC1 individual position and velocity errors, illustrated in figure 5.3, is included as a worst case representation of error profiles. Table 5.3 present fast block power ranking comparison. Additional fast trajectory IC1 data is included in appendix A. More detailed data representation is contained in [46].

5.4.1 Simulated Open-loop Torque Evaluation

The task of real-time reevaluation of dynamic models for robot control is complicated by the model and parameter specific nature of the current efficient dynamics algorithms ([38],[44]). To provide an estimate of the effects of Tarn's model on the conclusions of chapter four his inertial parameters were coded into the general Lagrange-Euler algorithm. A series of open-loop torque plots were generated using the complete, full and diagonal inertia dynamical representations with gravity but without Coriolis and centrifugal terms. Identical configurations were employed in chapter four. Figure A.1 graphically compares torques generated using Tarn's model to profiles generated by the original inertial parameters. Analysis of those torques reveals that the basic relations between the level of reduction in dynamic completeness and output torque is analagous for both cases. Therefore, the effect of neglected feedforward loop dynamics on real-time computed-torque performance should be similar.

5.4.2 Real-time Torque Comparison

To enable real-time evaluations, the efficient representations of manipulator dynamics had to be altered to conform to Tarn's model. That task was accomplished without symbolically reevaluating the whole formulation for the Lagrange-Euler dynamics by modifying the constant generator subroutine [38]. Although not an exact representation of the

Tarn model, figure A.2 shows that the torque discrepancies over the test trajectories with this hybrid model are negligible.

Real-time reevaluation of dynamic models for robot control is conducted over the identical trajectories employed in chapter four. The algorithms were evaluated with actuator inertia modeling.

Analysis of figures A.3-4 validates the open-loop torque conclusions about improved modeling's impact on the evaluation of dynamics for robot control. Table 5.3 compares the power rankings for the two sets of inertial parameters. The reduction in coupled dynamics torque values produced by Tarn's inertial parameters increases the dominance of the diagonal dynamics. Erroneous modeling of the inertial parameters is not responsible for the inability of Lagrange-Euler manipulator modeling to accurately represent actual dynamics.

Figure 5.3 compares the diagonal dynamics performance of the two manipulator models. Due to the dominance of actuator inertias small link variations are negligible. Link 2 and 3 basic error profiles are independent of parameter selection. Utilization of Tarn's inertial parameters in the manipulator dynamics more accurately models the inertial and gravitational forces of the arm. Therefore, the improved parameter model is employed in all subsequent studies so that the effects of unmodeled

TABLE 5.3

ALGORITHM REAL-TIME POWER RANKING VARIATIONS DUE TO INERTIAL PARAMETERS FAST TRAJECTORY

	RAL			TARN		
DYNAMICS	FULL INERTIAL	BLOCK INERTIAL	DIAGONAL INERTIAL	FULL INERTIAL	BLOCK INERTIAL	DIAGONAL INERTIAL
IC0						
JOINT 1	0.01	1.12	2.32	0.04	0.23	5.83
JOINT 2	1.58	0.38	3.42	0.14	0.06	4.28
JOINT 3	0.03	0.80	2.72	0.14	0.15	3.02
JOINT 4	0.71	0.41	0.30	0.38	0.53	0.36
JOINT 5	0.01	0.62	1.01	0.15	0.28	0.65
JOINT 6	0.42	0.77	0.05	0.68	0.10	0.64
IC1						
JOINT 1	0.89	0.23	1.99	0.02	0.27	5.37
JOINT 2	0.16	0.07	4.15	1.91	1.60	3.14
JOINT 3	0.10	0.28	2.34	0.55	0.96	2.63
JOINT 4	0.56	0.23	0.23	0.05	0.36	0.33
JOINT 5	0.00	1.37	1.59	0.13	0.51	1.09
JOINT 6	0.69	0.19	0.01	0.57	0.10	0.52
IC2						
JOINT 1	0.87	1.06	0.24	0.82	1.07	0.93
JOINT 2	0.25	0.31	2.97	0.96	1.13	1.94
JOINT 3	1.20	1.21	1.74	0.52	0.50	1.26
JOINT 4	0.91	0.16	0.03	0.95	0.27	0.54
JOINT 5	1.80	0.46	0.91	0.83	0.50	0.22
JOINT 6	0.23	0.42	0.07	0.05	0.19	0.12

Power rankings illustrate relative performance by scaling and summation of the normalized peak and final position and velocity errors produced by different algorithms over identical trajectories. Power rankings range from zero to ten with the best performing algorithm annotated by the highest ranking. For additional information refer to table 4.2

forces may be better understood.

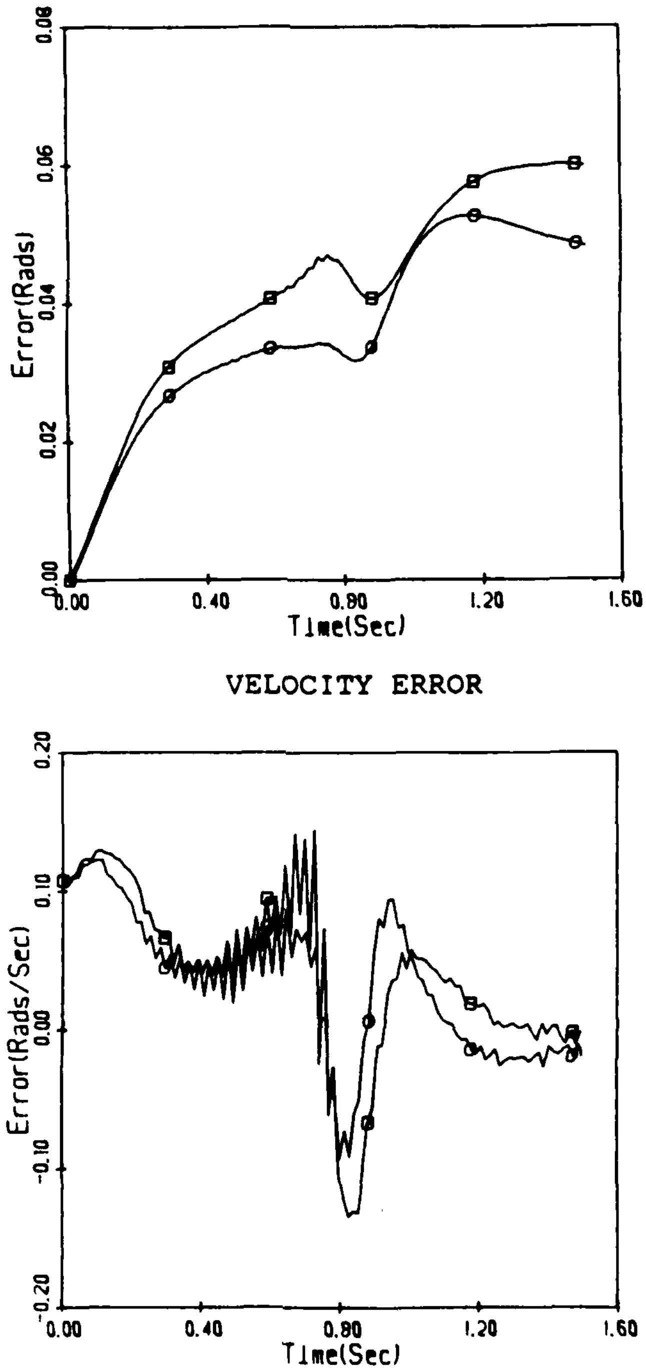
5.5 Unmodeled Force Compensation Evaluation

Real-time evaluation of the computed-torque performance improvement potential of friction compensation and feedback techniques is conducted over the identical trajectories employed in chapter four. To reduce strain on the manipulator due to vibrations the slow trajectory test is not averaged. The hybrid efficient diagonal dynamics presented in the last section are utilized. Listings of the PDP assembly language computed-torque algorithms are contained in [39].

Incorporation of static friction compensation into the feedforward loop by utilization of a nonlinear switching function reduces initial trajectory tracking errors but creates severe vibration after the midpoint. Feedback compensation techniques produce the advantages of friction compensation without the drawbacks. A computed-torque control law with a PID feedback loop produces trajectory tracking accuracy sufficient for control of high speed manipulator gross motion.

The comparison of fast trajectory IC1 individual joint position and velocity error for the uncompensated and compensated cases is illustrated in figure 5.4. Table 5.4 present power ranking comparisons. Detailed representation of additional data used in this analysis is contained in [46].

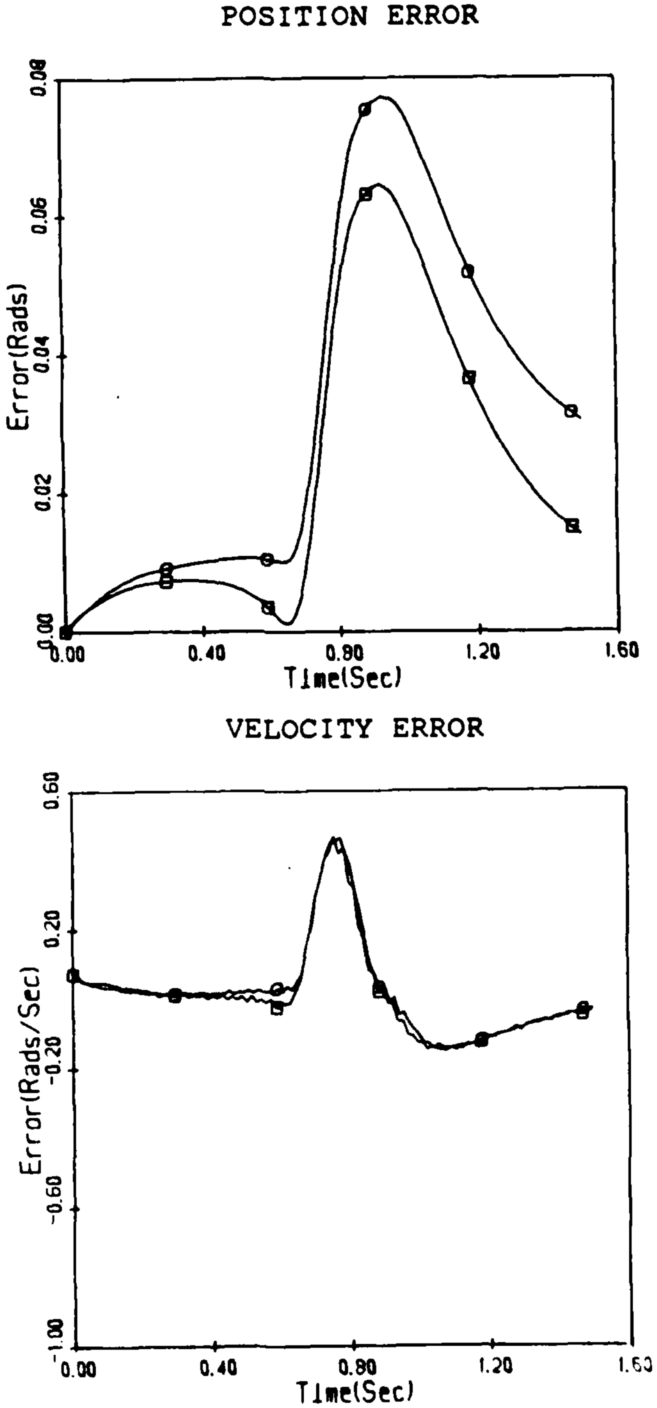
PERFORMANCE COMPARISON
DIAGONAL DYNAMICS



Refer to Table 5.1b for symbol definition

Figure 5.3a Joint 1 Fast IC1 Inertial Parameter Comparison

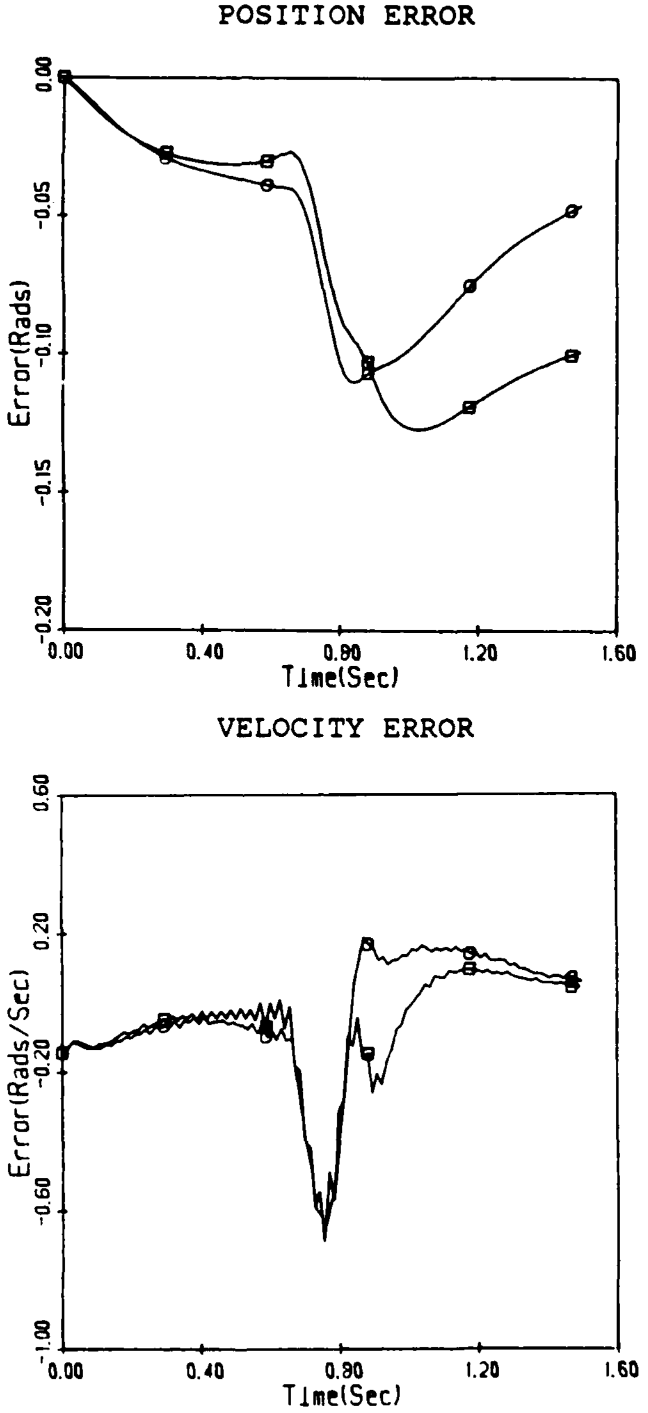
PERFORMANCE COMPARISON
DIAGONAL DYNAMICS



Refer to Table 5.1b for symbol definition

Figure 5.3b Joint 2 Fast IC1 Inertial Parameter Comparison

PERFORMANCE COMPARISON
DIAGONAL DYNAMICS

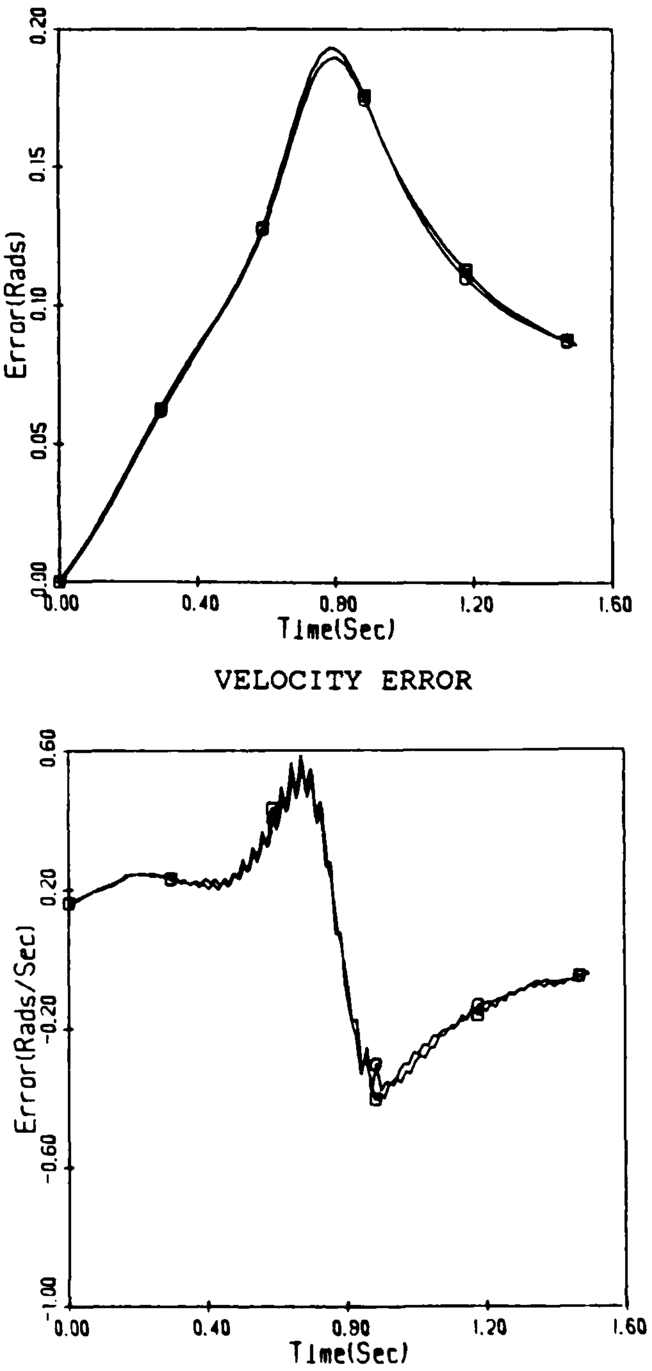

POSITION ERROR

VELOCITY ERROR

Refer to Table 5.1b for symbol definition

Figure 5.3c Joint 3 Fast IC1 Inertial Parameter Comparison

PERFORMANCE COMPARISON
DIAGONAL DYNAMICS



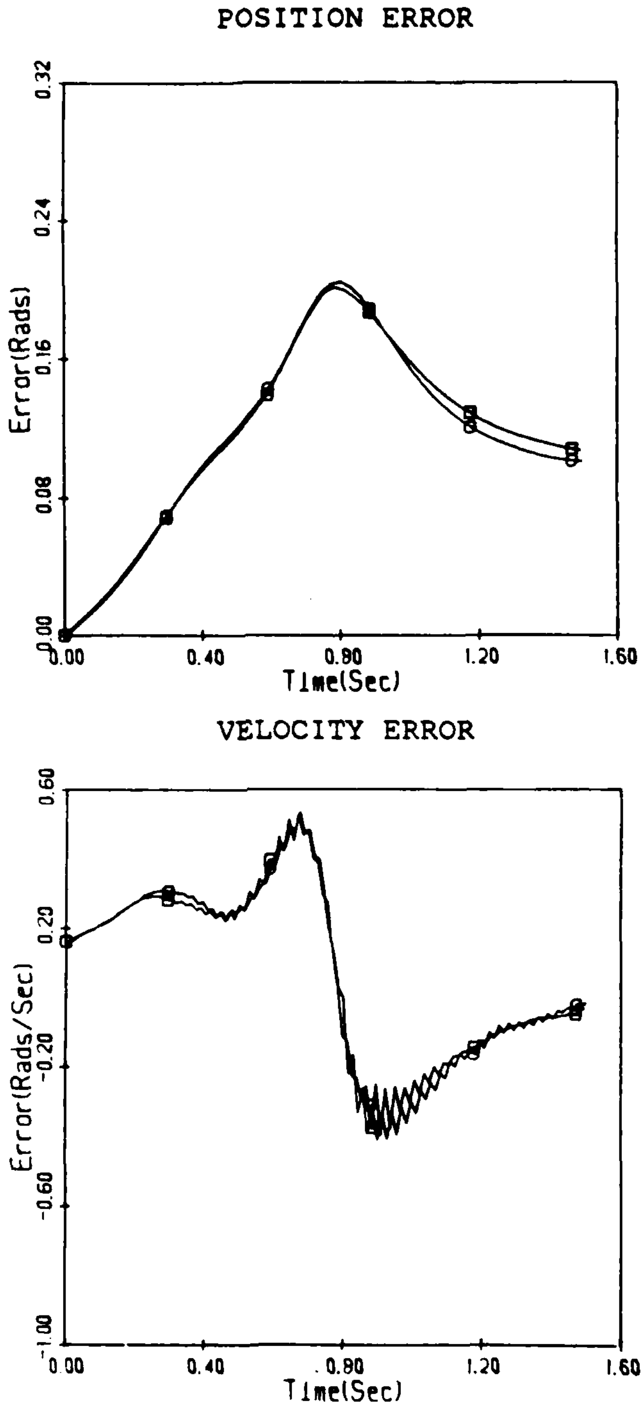
POSITION ERROR

VELOCITY ERROR

Refer to Table 5.1b for symbol definition

Figure 5.3d Joint 4 Fast IC1 Inertial Parameter Comparison

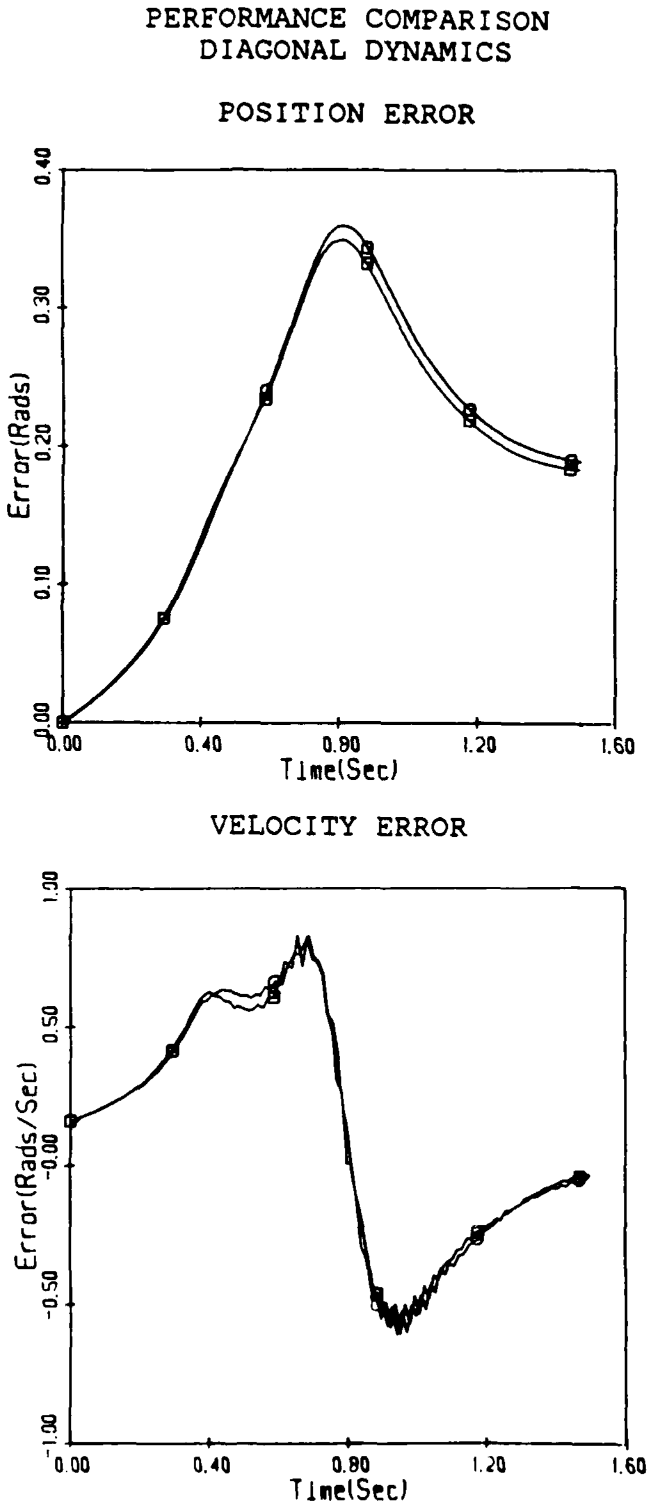
PERFORMANCE COMPARISON
DIAGONAL DYNAMICS


POSITION ERROR

VELOCITY ERROR

Refer to Table 5.1b for symbol definition

Figure 5.3e Joint 5 Fast IC1 Inertial Parameter Comparison



Refer to Table 5.1b for symbol definition

Figure 5.3f Joint 6 Fast IC1 Inertial Parameter Comparison

5.5.1 Static Friction Compensation

Without the friction compensation incorporated into equation 5.1 the nominal computed-torque position error profiles always showed each joint lagging the desired trajectory. With feedforward friction compensation large link error profiles display both lead and lag errors. Consequently, position error curves are a closer approximation of the diagonal dynamics simulation error curves. The general trend of the velocity error profiles is unaltered by modeling the friction. The shape of the error profiles is still initial position dependent. Friction compensation large link accuracy improvements are concentrated in the first half of the trajectory. Tracking improvements peak in joint 6 and are minimum for joint 2.

Improvements in small link accuracy are independent of initial position due to inertia actuator dominance. Unlike the large link case, friction compensation doesn't produce overshoot in small link error profiles. The improvement in joint 6 peak and final position errors are the most dramatic. Joint 6 maximum error is reduced by 0.09 and 0.04 radians respectively. The tradeoff for this improvement is increased vibration after the trajectory midpoint.

The joint stiction and 14ms sampling period combine with low nominal torque values to cause the nonlinear friction compensator to produce a series of bipolar torque steps. When applied to the manipulator that oscillatory input produces link vibration. The effect of those vibrations is greatest in the last link but best visualized in link 3. In general, for large links the increased velocity error offsets the improvements in position accuracy producing a lower power ranking than the original controller. Small link position improvements are so significant that their power rank enhancement can not be offset by the increased velocity errors.

5.5.2 Doubled Pole PD Feedback Loop

Figure 5.2 illuminated the tracking accuracy improvement potential of the doubled pole computed-torque technique represented by equation 5.2. Table 5.4 shows that performance improvements from increasing the frequency response of the computed-torque PD feedback loop are superior to the original loop with friction compensation (equation 5.1). The penalty of increased vibration is no longer incurred. All position errors lag. For large links the increased feedback gains deliver superior performance over the whole trajectory. The level of small link accuracy produced by the friction compensated algorithm is superior to that of the doubled pole PD loop for the first half second. That advantage rapidly disappears. Large link maximum final position error is within the target range of one degree. However, only joint 1 maximum peak position error is under three degrees. Small link maximum final position error is twenty-five percent of that produced by the uncompensated computed-torque control law. Unfortunately, small link position errors are still outside the target range.

5.5.3 PID Feedback Loop

By employing the computed-torque/PID control law of equation 5.3 maximum final position error for all links is held within the one degree target range. Average final position errors are half the maximum. Average peak position errors are under three degrees for the first five joints. However, only for links 1, 4 and 5 is maximum peak position error within the target range. Motion against gravity along the fast trajectory still produces unacceptably large link 2 and 3 peak errors. The penalties for this performance enhancement are trajectory overshoot and higher velocity error after the midpoint. In all cases that resulted in the final position overshooting the desired endpoint. The shape of the error profiles is identical for all the small joints. Once again the largest errors and the greatest improvement are in the last link.

The input torques were always below the limits of actuator saturation, eliminating the integrator windup compensation necessary in [21]. The non-optimal nature of the PID gains causes velocity error increases that allow the doubled pole PD loop to have a higher power ranking over the fast trajectories. Over the slow trajectory that discrepancy is less noticeable. By adjusting the gains for trajectory shape and speed those problems should be avoided.

TABLE 5.4

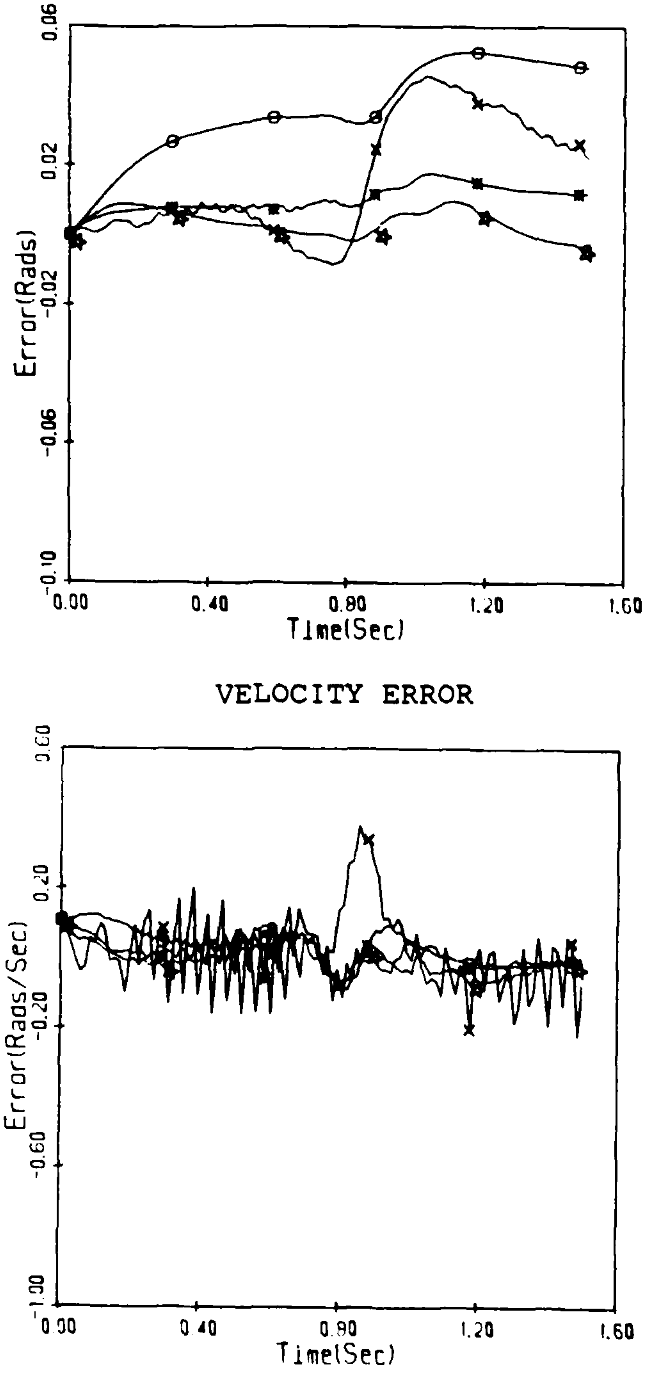
ALGORITHM REAL-TIME POWER RANKING VARIATIONS DUE TO COMPENSATION TECHNIQUES FAST TRAJECTORY

COMPENSATION	NONE	FRICTION	DOUBLED POLE PD	PID
IC0				
JOINT 1	2.96	2.18	7.44	8.29
JOINT 2	1.48	0.08	5.30	5.88
JOINT 3	1.93	0.46	5.27	4.75
JOINT 4	0.72	2.37	6.87	6.49
JOINT 5	1.60	2.33	7.31	7.68
JOINT 6	1.61	4.06	7.06	7.08
IC1				
JOINT 1	3.01	2.02	7.40	8.82
JOINT 2	1.63	3.32	4.93	3.08
JOINT 3	0.77	1.59	5.45	4.10
JOINT 4	1.00	2.48	7.16	6.80
JOINT 5	1.78	2.39	7.43	7.37
JOINT 6	0.71	5.73	7.22	5.73
IC2				
JOINT 1	1.02	1.75	5.40	5.85
JOINT 2	0.12	3.94	5.12	2.56
JOINT 3	1.47	1.47	5.98	5.95
JOINT 4	0.65	2.94	7.05	6.87
JOINT 5	0.54	4.09	6.84	6.84
JOINT 6	0.28	4.90	7.35	5.56

Power rankings illustrate relative performance by scaling and summation of the normalized peak and final position and velocity errors produced by different algorithms over identical trajectories. Power rankings range from zero to ten with the best performing algorithm annotated by the highest ranking. For additional information refer to table 4.2

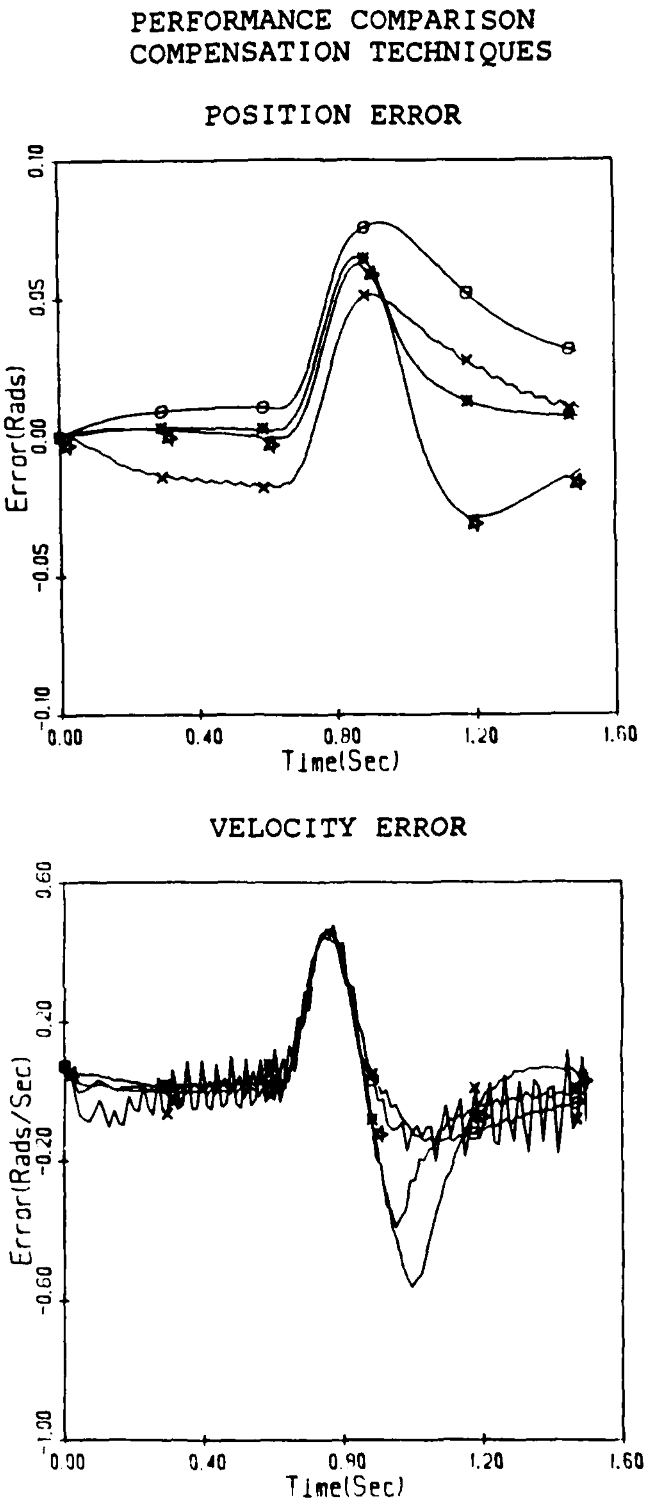
PERFORMANCE COMPARISON
COMPENSATION TECHNIQUES

Refer to Table 5.1b for symbol definition


POSITION ERROR

VELOCITY ERROR

Figure 5.4a Joint 1 Fast IC1 Errors


PERFORMANCE COMPARISON
COMPENSATION TECHNIQUES

POSITION ERROR

VELOCITY ERROR

Refer to Table 5.1b for symbol definition

Figure 5.4b Joint 2 Fast IC1 Errors

PERFORMANCE COMPARISON
COMPENSATION TECHNIQUES

Refer to Table 5.1b for symbol definition

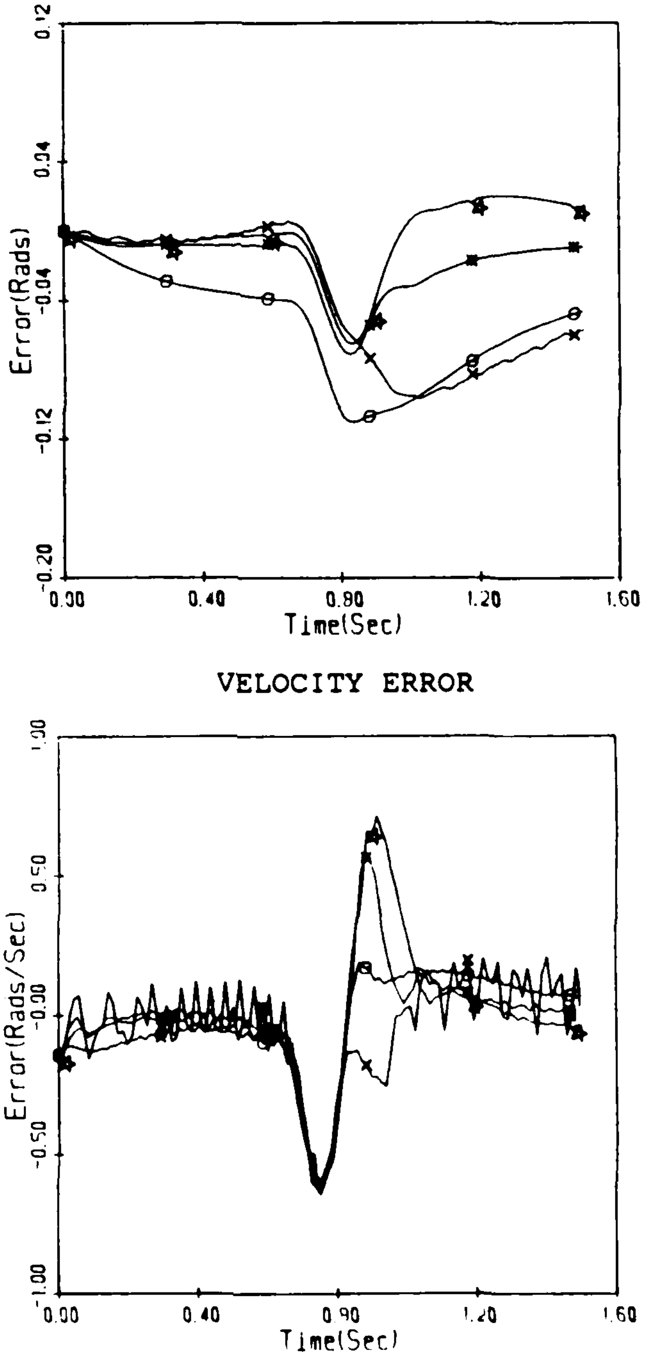
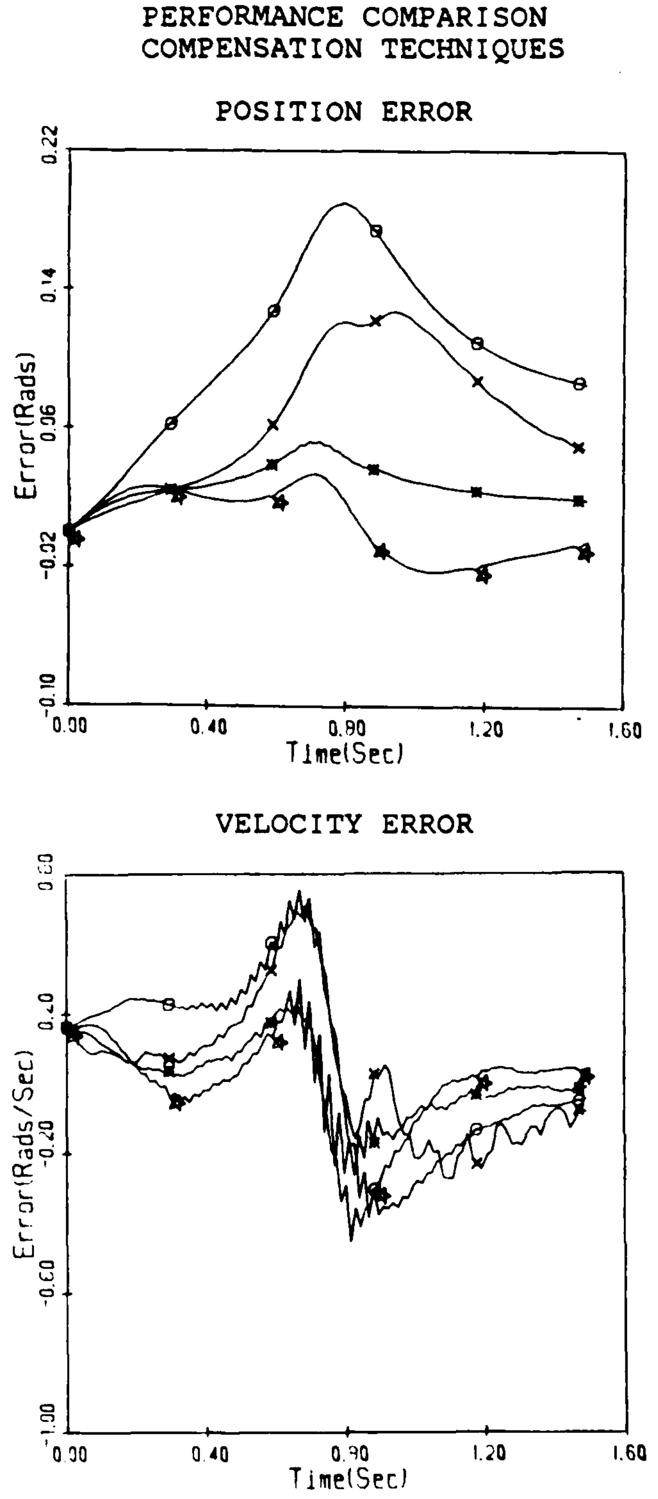


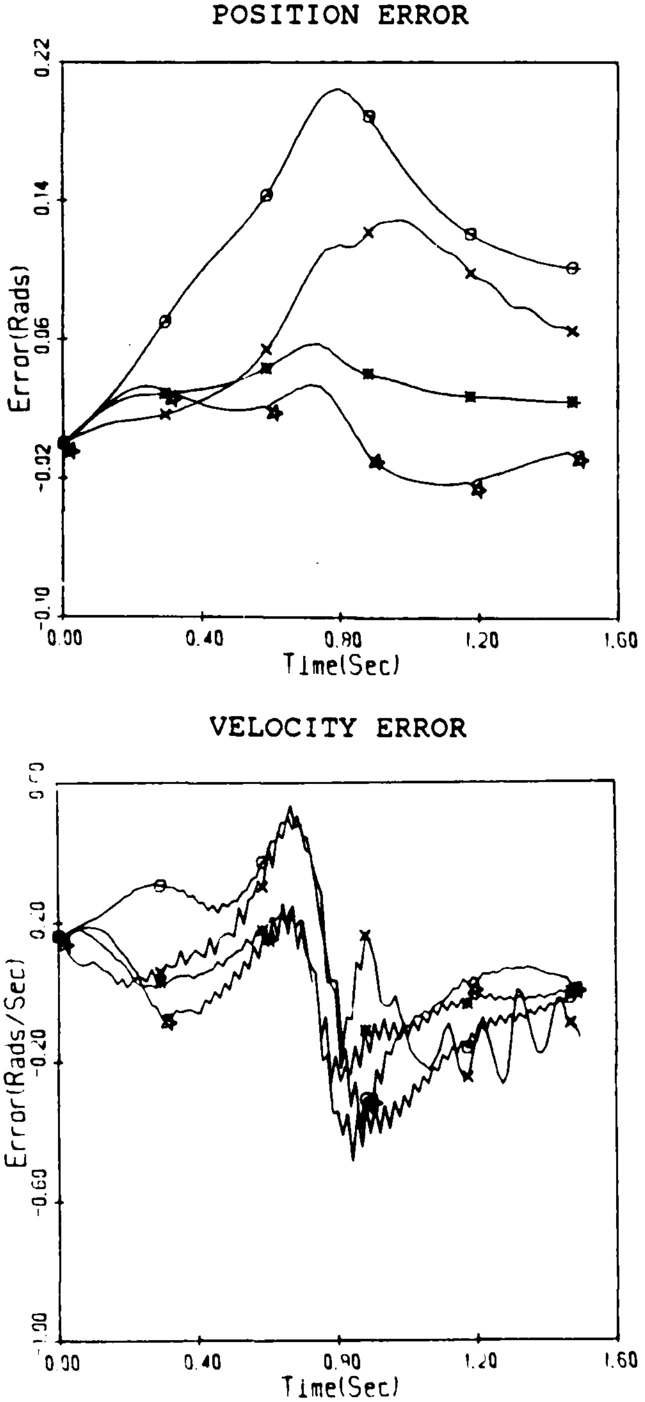
Figure 5.4c Joint 3 Fast IC1 Errors



Refer to Table 5.1b for symbol definition

Figure 5.4d Joint 4 Fast IC1 Errors

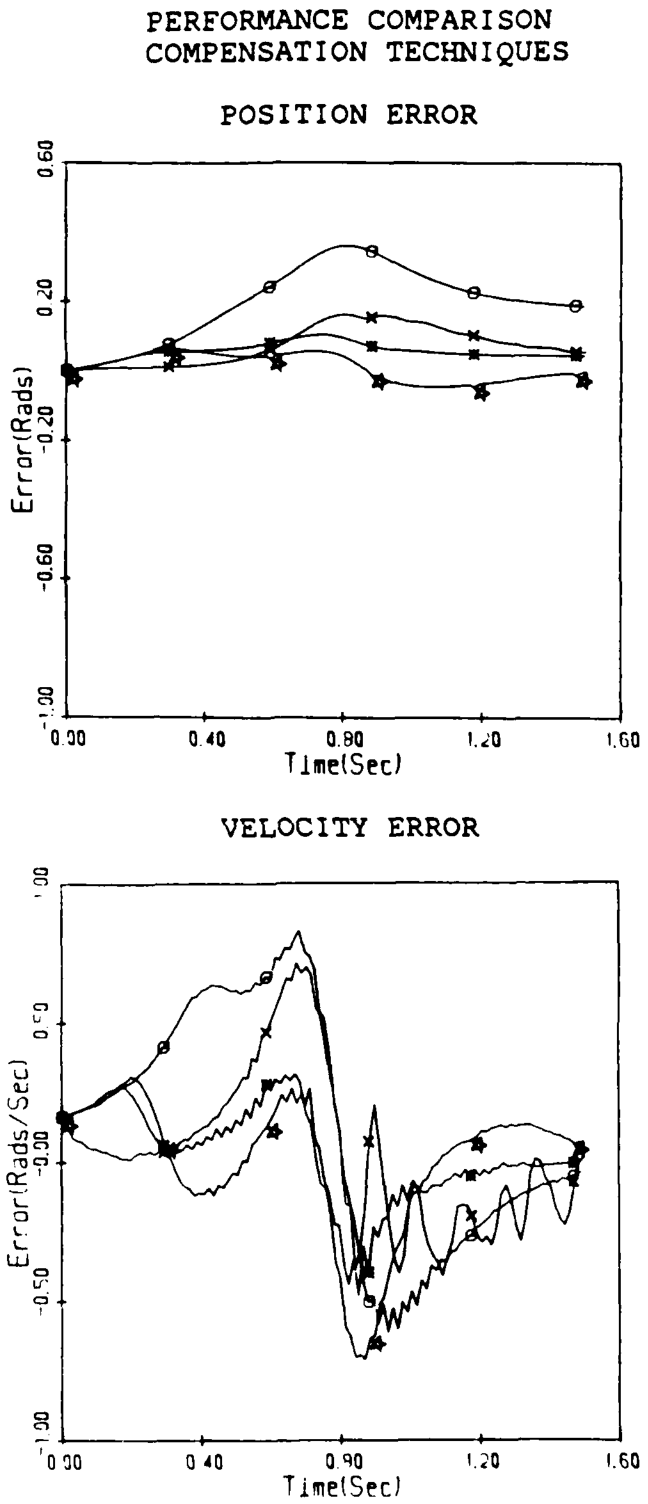
PERFORMANCE COMPARISON
COMPENSATION TECHNIQUES


POSITION ERROR

VELOCITY ERROR

Refer to Table 5.1b for symbol definition

Figure 5.4e Joint 5 Fast IC1 Errors



Refer to Table 5.1b for symbol definition

Figure 5.4f Joint 6 Fast IC1 Errors

5.6 Discussion

Robotic gross motion control research can be categorized into three main areas [84]:

1. joint motion control,
2. resolved motion control, and
3. adaptive control.

The additions to the real-time robotic control performance baseline produced by this chapter are applicable to all three categories. Information about feedforward dynamics can be applied to all joint and resolved motion dynamics based controllers. All groups employ some form of feedback control scheme whose real-time application will benefit directly from the lessons learned here.

The utilization of more accurate manipulator inertial parameters does not alter the conclusions of the previous chapter. Uncoupled feedforward dynamics still produce the best overall performance. More accurate knowledge of inertial parameters does improve algorithm performance, but not dramatically. Those results can be largely attributed to the dominance of the actuator inertias which were identical for both sets of inertial parameters. Accurate information about actuator inertias is the most important component in accurate modeling of highly geared manipulators.

The improvement in tracking accuracy from utilization of a nonlinear friction function in the feedforward loop is not worth the cost in increased vibration. Forces unmodeled by feedforward diagonal dynamics can be treated as a disturbance to the feedback loop. A more robust feedback loop produces the advantages of friction compensation without the cost of severe vibration. A PD loop reduces tracking errors without overshoot. By adding an integrator to the feedback loop the final errors can be reduced to within the feasibility sphere of one degree.

A real-time evaluation of computed-torque technique effectiveness in controlling the large links of the CMU direct-drive arm has recently been conducted [34]. The PD poles for that study are critically damped with Kp = sqrt(Kv). Sampling time was two milliseconds. Velocity gain was experimentally determined as eighty percent of the value that caused each individual joint to vibrate. The dynamics are accurately known and the role of unmodeled dynamic forces should be minimum. For the first 1.5 seconds the reference trajectories are very similar to those shown in figure 4.1 permitting valid comparison of computed-torque performance on direct drive and highly geared manipulators.

The maximum position errors produced on joints 1 and 2 of the low friction direct drive arm with optimized gains [34] are greater than those produced by by application of the uncompensated computed-torque control law (equation

4.1) to the PUMA. Utilization of the feedback compensated control laws of equations 5.2 and 5.3 magnifies that advantage. Only for joint 3, where the lack of gravitational forces allow more accurate control, was direct drive arm performance superior. The assumption that usage of direct drive arms would permit improved performance is clearly false.

Information from a study of feedforward controllers conducted at MIT [3] suggests that the effect of unmodeled forces become dominant as link inertia decreases, even for direct drive arms. Therefore, PUMA small link results are applicable to direct drive arms.

PD and PID feedback loops are utilized to control most industrial manipulators [61] so their ability to reject the disturbances of unmodeled dynamics is not suprising. In industrial applications that rejection ability is limited to slow motions were ignorance of dynamic forces is rendered harmless by their low values. The knowledge of dynamic forces provided by the computed-torque technique reduces nonlinear effects and adapts the feedback gains to manipulator configuration and task. The result is a more efficient and complaint controller which is not restricted to slow motions.

Further experimental data will allow educated selection of the cost criterion matrices so that optimal control techniques could be applied to calculate the PID gains [65]. Improvements in control computer architecture will produce greater performance via increased sampling rates. Torque sensors can be mounted in the joints and the information utilized to eliminate the effects of friction [80]. Guidelines for design of manipulators with decoupled and configuration independent inertia matrices have been developed [110]. The fusion of these electrical and mechanical developments will allow production of robotic arms with the desirable characteristics of direct drive arms without the requirement of extensive dynamics computation and restricted applications due to limited range of motion and payload.

5.7 Summary

Another major contribution to the real-time robotic control data base has been completed. The results from evaluation of unmodeled dynamics compensation techniques are summarized in table 5.5.

The implementation feasibility and performance improvement potential of feedforward and feedback compensation of dynamically based manipulator control techniques have been clearly illustrated. The feedback control information added to the control database is applicable to all proposed feedback control schemes [56]. Implementation of computed-torque control with suboptimal gains on a highly geared manipulator produces trajectory tracking performance superior to optimum gain application on a direct drive arm [34].

The evaluation of techniques for compensation of unmodeled manipulator dynamics has revealed that:

1. utilization of more accurate inertial parameters does not alter the conclusions of chapter four,
2. utilization of more accurate inertial parameters does not significantly improve computed-torque trajectory tracking ability,
3. friction c.mpensation by a nonlinear switching function in the feedforward loop produces unacceptable performance,
4. forces unmodeled by feedforward diagonal dynamics can be treated as disturbances to the feedback loop, and
5. tracking accuracy sufficient for gross motion control of a highly geared manipulator operating at the edge of its performance envelope is achievable without additional instrumentation.

The accuracy of the manipulator controller has been improved to the point that errors in calibration now become significant. Improvements in calibration accuracy and/or knowledge of the calibration uncertainty is necessary for improved performance and successful integration with the additional components that constitute a hierarchically controlled intelligent machine. A theoretical investigation of calibration uncertainty is the subject of the next chapter.

TABLE 5.5

ALGORITHM REAL-TIME POWER RANKING
OVERALL COMPARISON

COMPENSATION	NONE	FRICTION	DOUBLED POLE PD	PID
SLOW	2.44	2.65	6.22	7.34
FAST	1.29	2.67	6.48	6.09
OVERALL	1.86	2.66	6.35	6.72

Slow and Fast values represent power ranking data averaged over all initial conditions and joints. Overall values average Slow and Fast data.

CHAPTER 6

CALIBRATION UNCERTAINTY

6.1 Introduction

The control technique evaluations of the previous two chapters were dependent on the repeatability of the manipulator control electronics, not the ability to perform extremely accurate calibrations. By subjecting all algorithms to identical test configurations without recalibrating the manipulator the small errors between modeled and actual joint position were rendered irrelevant. However, application of those techniques to a manipulator in an integrated work cell environment does require extremely accurate knowledge of the cartesian position. The level of accuracy depends on the calibration procedure which aligns the manipulator with the external environment. The precision of the calibration procedure is dependent on a set of idealized assumptions and parameters. Variations in those parameters produced by manufacturers' tolerances combine with positioning system imprecision to create a level of uncertainty in the calibrated position of the manipulator.

A fundamental assumption of the Computed-torque, feedforward ([3],[35]), and nonlinear feedback techniques ([10],[20],[95]) is that the dynamical parameters are well known. Chapter four demonstrated that those techniques are stable in the presence of parameter uncertainties, but efficacy decreases. Evaluation of end-effector tracking accuracy degradation due to uncertainty in inertial and load parameters is the next step in dynamical control law research. A prerequisite for those evaluations is the ability to separate the calibration and dynamically based uncertainties.

An intelligent machine will have the ability to select the appropriate control algorithm for a certain task. Intelligent control algorithm decision making requires knowledge of the operational environment. A necessary component in that knowledge base is a measure of the uncertainty in the calibrated position of the manipulator.

In this chapter a significant contribution to the field of hierarchical intelligent control research is achieved by development of the theoretical basis for employment of the Entropy function as a measure of the calibration uncertainties. The theory is developed for the general case and then applied to a PUMA manipulator. The Entropy function provides an uncertainty measure consistent with the hierarchical control architecture proposed by Valavanis and Saridis ([85-6],[105]) while providing the prerequisite information needed for continued dynamical based control research.

6.2 Problem Statement

In an autonomous intelligent work cell the standard industrial practice of teaching trajectory position to the manipulator after calibration is abandoned. Knowledge of joint or end-effector position relative to an external reference, not an internal one, is now necessary. The ability to calibrate the manipulator relative to an external frame is clouded by uncertainties.

The primary sources of calibration imprecision are uncertain knowledge of [108]:

1. joint encoder offsets,
2. relative orientation of consecutive axes, and
3. kinematic parameters

Those uncertainties combine with the unmodeled real world effects of joint compliance, backlash, gear transmission error, and control system position imprecision to produce calibration uncertainties.

Current calibration research is centered around elimination of these uncertainties by utilization of external instrumentation ([19],[60],[92],[108]). Better knowledge of those parameters reduces, but does not eliminate calibration uncertainties. This research is not concerned with the explicit reduction of individual parameter uncertainty but rather with a calculation of overall calibration uncertainty. Knowledge of calibration uncertainty would allow for its compensation.

In chapter two the two major techniques for calculation of uncertainty were reviewed. Only the probabilistic approach fulfills our requirement to provide an uncertainty measure consistent with the other information sources presented to the hierarchy proposed by Valavanis and Saridis ([85-6],[105]) while providing the cabability to learn, and therefore compensate for the uncertain nature of the real world [83].

6.3 Method Of Approach

The theoretical basis for utilization of the entropy function as a measure of calibration uncertainty is developed in stages. The general theory for joint space uncertainty calculation is developed first. Since uncertainties in the forward kinematics combine with the joint space uncertainties to produce uncertainty in calibrated end-effector position the theoretical development is expanded to allow calculation of cartesian space calibration uncertainty. As a practical demonstration the generalized theories will be applied to calculation of the uncertainty produced in PUMA manipulator calibration. A numerical example is presented. The study concludes with discussions of applications areas.

6.4 General Theoretical Development

Modern industrial manipulators are controlled at the joint level. Before the manipulator can be utilized each of those joints must be calibrated. The uncertainty produced by that calibration process can be subdivided into joint and cartesian space sources.

6.4.1 Joint Space Uncertainty

Consider a joint whose calibration error is constrained into an interval, D, of the joint angular space with a measurement accuracy of Δ. Then the error in calibrated position of each joint can be represented by a discrete probability distribution. Discrete probability distributions are employed due to the finite resolution of the joint position measurement devices. The discrete joint calibration error probability distribution is:

$$P(Q=q_i) = P(q_i) \geq 0 \quad (i = 1,\ldots n) \tag{6.1}$$

$$\sum_{i=1}^{n} P(q_i) = 1 \tag{6.2}$$

Where:

Q = The random variable of joint error

q_i = The discrete values of Q

n = $\frac{D}{\Delta}$, the number of discrete values of q_i

$P(Q=q_i)$ = Probability that Q equals q_i

The level of uncertainty in the calibration of that joint can then be expressed by the following Entropy function:

$$H(Q) = -\sum_{i=1}^{n} P(q_i) \log_2 P(q_i) \tag{6.3}$$

Arbitrary positioning of the end-effector in three dimensional space requires at least six degrees of freedom. Therefore the manipulators of interest must have N joints where N $\geq$ 6. For an N joint manipulator the overall calibration error could be dependent on the uncertainty of each joint. In that case the overall manipulator calibration error may be expressed in the N joint space as:

$$P(C_Q) = P(Q_1 = q_{1_i}, Q_2 = q_{2_j}, \ldots, Q_N = q_{N_k})$$

$$= P(q_{1_i}, q_{2_j}, \ldots, q_{N_k}) \geq 0 \tag{6.4}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \ldots \sum_{k=1}^{p} P(q_{1_i}, q_{2_j}, \ldots, q_{N_k}) = 1 \tag{6.5}$$

Where:

Q_a = Random variable for each joint (a=1,...N)

$q_{a_{i,j,\ldots,k}}$ = Discrete values of Q_a

i,j...k = Number of discrete values of $Q_1, Q_2 \ldots Q_N$

$P(q_{1_i}, q_{2_j}, \ldots, q_{N_k})$ = Probability $\{Q_1=q_{1_i}, Q_2=q_{2_j}, \ldots, Q_N=q_{N_k}\}$

The total joint space calibration uncertainty can then be expressed as:

$$H(Q_1, Q_2, \ldots, Q_N)$$

$$= - \sum_{i=1}^{n} \sum_{j=1}^{m} \ldots \sum_{k=1}^{p} P(q_{1_i}, q_{2_j}, \ldots, q_{N_k}) \log_2 P(q_{1_i}, q_{2_j}, \ldots, q_{N_k}) \tag{6.6}$$

6.4.2 Cartesian Space Uncertainty

Although the manipulator is controlled at the joint level the position of the manipulator end-effector, not that of the individual joints is generally the point of interest. The cartesian position of the end-effector can be represented by a vector defined as:

$$X = \begin{bmatrix} P \\ \Omega \end{bmatrix} = [X_1, X_2, \ldots, X_6]^T \qquad (6.7)$$

Where:

X = (6x1) Vector of end-effector position in cartesian space.

P = (3x1) sub-vector containing X, Y, Z position information.

Ω = (3x1) sub-vector containing Euler angles.

The cartesian and joint positions can be related by a series of homogeneous coordinate transformations which utilize the kinematic parameters of the links to relate the joint angles to cartesian position [84]. Therefore -artesian space calibration error is a function of joint space calibration errors and variations in kinematic parameters. If there are L kinematic parameters then the cartesian calibration error can be represented by:

$$E = [e_1, e_2 \ldots e_6]^T \qquad (6.8)$$

Where:

E = (6x1) vector of end-effector cartesian calibration error.

$$e_i = f_i(q_1, q_2, \ldots, q_N, k_1, k_2, \ldots, k_L) \qquad (6.9)$$

Where:

k_i = Kinematic parameter errors (i=1,.., L)

If the cartesian calibration error probability is expressed as:

$$P(E) = P(E_1=e_{1_i},\ E_2=e_{2_j},\ldots,\ E_6 = e_{6_k}) \tag{6.10}$$

$$= P(e_{1_i},\ e_{2_j},\ldots,\ e_{6_k}) \geq 0$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \ . \ . \ . \sum_{k=1}^{p} P(e_{1_i},\ e_{2_j},\ldots,\ e_{6_k}) = 1 \tag{6.11}$$

Where:

E_a = random cartesian calibration error variables (a=1,..,6)

$e_{ai,j,..,k}$ = discrete value of E_a

i,j,..,k = number of discrete values of E_a

$P(e_{1_i},\ e_{2_j},\ldots,\ e_{6_k})$ = Probability$\{E_1=e_{1_i},\ E_2=e_{2_j}\ldots,\ E_6=e_{6_k}\}$

The corresponding uncertainty measure is:

$$H(C_x) = H(E_1,\ E_2,\ldots,\ E_6)$$

$$= -\sum_{i=1}^{n} \sum_{j=1}^{m},\ldots, \sum_{k=1}^{p} P(e_{1_i},\ e_{2_j},\ldots,\ e_{6_k}) \log_2 P(e_{1_i},\ e_{2_j},\ldots,e_{6_k}) \tag{6.12}$$

To provide a better insight into the general theory the equations for calculation of calibration uncertainty are applied to a PUMA manipulator.

6.5 PUMA Case Study

The PUMA robot arm is a six degree of freedom revolute manipulator. PUMA joint space calibration error probability can be expressed as:

$$P(C_Q) = P(q_{1_i},\ q_{2_j},\ldots,\ q_{6_k}) = P(Q_1,\ Q_2,\ldots,Q_6) \tag{6.13}$$

Bayes theorem [23] allows equation 6.13 to be rewritten as:

$$P(C_Q) = P(Q_1)P(Q_2|Q_1)P(Q_3|Q_2Q_1)\ldots P(Q_6|Q_5\ldots Q_1) \tag{6.14}$$

The PUMA has the following characteristics that allow equation 6.14 to be simplified:

1. the calibration procedure is independent for the first four joints,
2. joint five calibration error is dependent on joint four error, and
3. joint 6 calibration error is dependent on the error of both joint 4 and 5.

Those characteristics allow equation 6.14 to be reduced to:

$$P(C_Q) = P(Q_1)P(Q_2)P(Q_3)P(Q_4)P(Q_5|Q_4)P(Q_6|Q_5Q_4) \tag{6.15}$$

The corresponding calibration uncertainty measure has become:

$$H(C_Q) = H(Q_1)+H(Q_2)+H(Q_3)+H(Q_4)+H(Q_5|Q_4)+H(Q_6|Q_5Q_4) \tag{6.16}$$

The sources of uncertainty in calibrated joint angular position are:

1. joint encoder offset, (J),
2. imprecision of positioning control system, (P), and
3. idealized assumptions about the effects of backlash and gear transmission error, (B).

Therefore the joint calibration errors may be written as:

$$q_i = f(J_i, P_i, B_i) \tag{6.17}$$

with the corresponding changes in uncertainty representation.

For the kinematic model of the PUMA employed in the RAL the homogeneous transformation matrices are illustrated in ([45],[84]). Those transformations can be concatenated to produce the end-effector matrix. The end-effector matrix contains all the information necessary to determine the position and orientation of the end-effector. Joint and cartesian space calibration uncertainty are related by the end-effector matrix.

There are five kinematic parameters associated with the end-effector matrix of the PUMA (a2,d2,a3,d4,d6) [84] so that:

$$[k_1, k_2, \ldots, k_5] = \text{error in } a_2, d_2, a_3, d_4, d_6 \text{ respectively}$$

By analyzing the explicit symbolic equations of the end-effector matrix [84] the functional dependence of the cartesian calibration errors can be reduced to:

$$e_1 = f(q_1 \ldots q_5, k_1 \ldots k_5) \quad (6.18)$$

$$e_2 = f(q_1 \ldots q_5, k_1 \ldots k_5) \quad (6.19)$$

$$e_3 = f(q_2 \ldots q_5, k_1, k_3 \ldots k_5) \quad (6.20)$$

$$e_4 \ldots e_6 = f(q_1 \ldots q_6) \quad (6.21)$$

Unfortunately the six cartesian calibration errors are all dependent so that equation 6.10 cannot be simplified in a manner similar to equation 6.15.

6.6 Numerical Example

As a check of the uncertainty calculation, assume that the calibration of each joint is exact. The individual joint space calibration error probability function is shown in figure 6.1. The resultant calibration entropy is:

$$H(Q_i) = -\sum_{i=1}^{n} P(q_i) \log_2 P(q_i) \tag{6.22}$$

$$H(C_Q) = -\sum_{i=1}^{6} H(Q_i) = \sum_{i=1}^{6} 0 = 0 \tag{6.23}$$

Equation 6.23 verifies the ability of the Entropy function to correctly represent the lack of calibration induced position uncertainty.

In the absence of other information the error in joint calibration could be represented by a uniform distribution bounded by experimentally determined limits as shown in figure 6.2. For a uniform distribution the probability is 1/n. Therefore the level of uncertainty is:

$$\begin{aligned} H(Q_i) &= -\sum_{j=1}^{n_i} \frac{1}{n_i} \log_2 \frac{1}{n_i} \\ &= \log_2 n_i \text{ bits} \end{aligned} \tag{6.24}$$

$$H(C_Q) = \sum_{i=1}^{6} \log_2 n_i \tag{6.25}$$

Experience with the PUMA shows that maximum joint calibration error is less than two degrees. Table 6.1 lists the experimentally determined error limits for joints 1-6. The number of discrete values for each joint error is determined by assuming a measurement resolution of 0.005 degrees. If

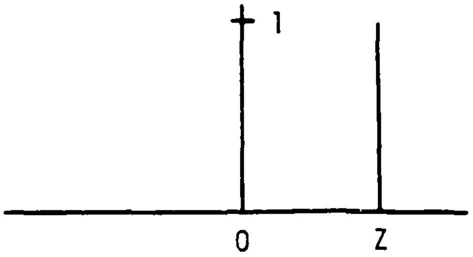


Figure 6.1 Calibration Error Probability Distribution With No Uncertainty

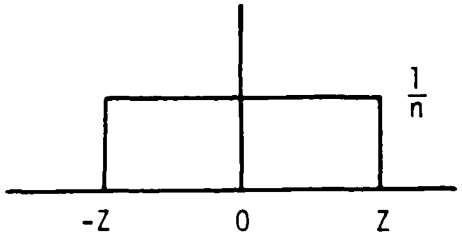


Figure 6.2 Calibration Error Probability Distribution With Maximum Uncertainty

TABLE 6.1

Experimental Calibration Error Data

Joint	$\|Z\|$ Degrees	n_i
1	1.5	300
2	1.5	300
3	3.0	600
4	1.0	200
5	0.5	100
6	1.5	300

the joint calibration error probabilities are assumed to be uniform then the calibration uncertainty can be calculated from equation 6.25 and the resultant value is 13.48 bits.

These examples highlight the ability of the Entropy function to convey calibration uncertainty information to the upper levels of an hierarchically intelligent machine. An overview of techniques for reduction of calibration uncertainty is presented next. Further analysis of methods for reducing the uncertainty levels is beyond the scope of this research.

6.7 Uncertainty Reduction

An overview of the reduction methods applicable to calibration uncertainties can be subdivided into off and on-line categories. Calibration uncertainty can be reduced off-line by utilization of sophisticated measurement devices [108] to obtain better knowledge of the kinematic and joint offset parameter error probability distributions. Tighter probability distributions reduce the Entropy of equation 6.12. The calibration uncertainty kinematic parameter functional dependence (equations 6.18-21) should be analyzed. Symbolic methods similar to those employed by Brooks [11] can be employed to accomplish that task. Knowledge of those relationships permits determination of the dominant players in calibration uncertainty and focuses reduction efforts on their uncertainty.

The effects of backlash, compliance and gear transmission error must be reduced on-line. An experimental setup similar to the one proposed by Foulloy and Kelley [19] permits information about the uncertainties produced by those phenomenon to be collected. Accelerometer data required to implement the feedback linearization scheme of Luo and Saridis [65] also provides uncertainty information in the form of errors between the calibrated and actual gravity normal position. Uncertainty information can be input to a stochastic learning algorithm that updates the calibration error probabilities and thus reduces the Entropy function associated with calibration uncertainties. Saridis and Blumberg have proposed such techniques for minimization of the Entropy associated with linguistic decision schema in a hierarchical intelligent machine [87].

6.8 Discussion

Knowledge of calibration uncertainty is a powerful tool for enabling intelligent selection of manipulator control algorithms. In an intelligent work cell environment the movement of the manipulator will be controlled by different control laws depending on task and environment. The difference may be as small as a change of gains in the controller or as radical as a switch to a completely new control law. The research of chapter's 4 and 5 demonstrated the feasibility of manipulator control by feedback linearization and application of linear control laws. By utilizing an optimal control law to control the linear system the cost of control associated with the penalty matrices can be treated as an Entropy.

All the information associated with selection of a particular control technique will be summed into one Entropy value. One of those cost will be the penalty for lack of precision produced by uncompensated uncertainties. When the Entropy from utilization of that control law exceeds the performance criterion determined by the upper levels of the hierarchy the control formulation will be altered.

In a robotic system with multiple forms of position sensing the hierarchy will strive to employ the most efficient instrumentation for the given task. Knowledge of the uncertainty in calibrated position is a vital input to that decision making process. In minimal uncertainty environments external sensor information may be unnecessary. In the presence of low uncertainty low cost range sensors, infra-red, may be sufficient to ascertain actual end-effector position. Maximum uncertainty would require utilization of the more costly full vision system to obtain the same information.

6.9 Summary

The Entropy function provides a measure of calibration uncertainty compatible with the other operational environment and algorithms performance criterion information sources provided to the upper levels of a hierarchically intelligent machine. Sources of joint and cartesian space calibration uncertainty were identified. The theoretical basis for calculation of calibration uncertainty by utilization of an Entropy function was developed. Techniques for Entropy minimization can be employed to reduce calibration uncertainties and consequently improve the performance of the manipulator control techniques previously evaluated. Application of calibration uncertainty calculation theory to a PUMA manipulator illustrated the advantages inherent in the employment of Entropy as a measure of calibration uncertainty in a hierarchical intelligent environment.

CHAPTER 7

CONCLUSIONS AND FUTURE RESEARCH

7.1 Summary Of Results

A dynamically based robotic manipulator controller performance baseline has been established by the creation and utilization of a hierarchical robotic evaluation environment. Analysis of baseline information has significantly reduced the search for a gross motion control scheme applicable to intelligent machines.

Creation of a hierarchical robotic evaluation environment provided an original solution to the problems that previously constrained real-time evaluation of modern manipulator control techniques. That solution was developed by integration of three major integrated components: a hierarchical manipulator control system, customized efficient algorithms for computation of manipulator dynamics, and software libraries that support simulation and real-time modern control algorithm performance evaluation.

The Hierarchical Robotic Evaluation Environment propels the RAL to the forefront of robotic manipulator modern control technique application research. The RAL Hierarchical Control System (RHCS) provides the framework for the investigation of numerous areas of robotic control research. The RAL Real-Time Robotics Algorithm Exerciser (R3AGE) permits evaluation of all proposed joint, resolved motion and adaptive control algorithms. Simplicity, power, and expandability make implementation of this environment an optimal solution for any institute in search of a real-time testing platform.

A major deficiency in robot control research has been insufficient experimental evaluation of proposed techniques. Utilization of the Hierarchical Robotic Evaluation Environment for real-time evaluation of dynamically based manipulator control has significantly reduced that deficiency by establishing a control performance baseline.

The Hierarchical Robotic Evaluation Environment was utilized to evaluate Lagrangian dynamics for robot control and investigate the performance improvement potential of techniques for compensation of unmodeled forces. The case studies were performed on a PUMA-600 manipulator controlled by various forms of the computed-torque technique. The generic nature of the computed-torque technique allows knowledge acquired from performance evaluations to be extended to all PUMA-600 manipulator control algorithms that employ dynamics based linearization and/or classical or state space designed feedback loops.

Although the validity of these results has only been proven for a PUMA-600 manipulator, they provide valuable insight into modern robotic control theory applications. The mechanical equivalence between the PUMA 560 and 600 should allow these results to be directly extended to PUMA 500 series manipulators.

The two pioneering case studies combined to establish the dynamically based PUMA controller real-time performance baseline. The major conclusions from those studies are:

1. control algorithm comparison studies employing the complete Lagrange or Newton-Euler models to simulate the PUMA manipulator produce invalid results,
2. the effects of Coriolis and centrifugal forces are negligible,
3. unmodeled forces cancel the benefits of inertial coupling displayed in the simulation study for all links except the fourth,
4. inclusion of reflected actuator inertias in the feedforward loop significantly enhances tracking accuracy especially for the small links,
5. gravity forces are significant and should be modeled in the feedforward loop,
6. diagonal inertial terms are significant and should be modeled in the feedforward loop,
7. utilization of more accurate inertial parameters does not significantly impact controller effectiveness,
8. friction compensation by a nonlinear switching function in the feedforward loop produces unacceptable performance,

9. forces unmodeled by feedforward diagonal dynamics and gravity can be treated as disturbances to the feedback loop, and

10. dynamically based control techniques have the potential to control high speed gross manipulator motion without additional sensor devices.

The tracking accuracy of the computed-torque/PID controller with suboptimal gains confirms the suitability of the LQ design approach of Luo and Saridis [65]. The ability to represent the optimal control penalty matrices as Entropy functions makes this control technique particularly attractive in a hierarchical intelligent control system.

A nonheuristic original solution to the problem of calculation of calibration uncertainty was developed. The theoretical basis for representation of joint and cartesian space calibration uncertainty by an Entropy function was established. That research provides the foundation for continued development of an intelligent hierarchically based controller. Armed with knowledge of calibration uncertainty the intelligent machine can select controllers appropriate for the level of position uncertainty.

7.2 Recommendations For Future Research

This research provides the foundation for continued research into development of control methods applicable for an intelligent machine. That foundation should be employed to expand the manipulator control database. The following studies are suggested:

1. Paul proposed that dynamics do not have to be updated at the same rate as the control law [32]. If that proposal is valid the performance enhancements produced by faster sampling times could be achieved without additional computational power. The validity of that proposal must be investigated. A relationship between PUMA dynamics update rate and controller effectiveness should be conducted.
2. The evaluation of computed-torque/PID control should be expanded to study the effect of the LQ design techniques proposed by Luo and Saridis [65]. Emphasis should be placed on determining the relationships between weighting matrices and manipulator performance.
3. More accurate representations of motor dynamics and gear-train friction should be developed and the impact of their feedforward modeling evaluated.
4. The knowledge gained from joint motion evaluations should be applied to resolved motion acceleration control so that a resolved motion performance baseline can be established. Efficient Jacobian inversion software ([45],[74]) permits real-time resolved motion testing under R3AGE.
5. Adaptive control techniques should be evaluated to determine if their performance is superior to dynamics based techniques. Evaluation of Dubowsky's Model

Reference Adaptive Control technique [16-17] is in progress.

6. The performance potential and implementation feasibility of the sensor based feedback linearization technique of Luo and Saridis [65] should be investigated.
7. The effects of end-effector load variations on algorithm performance should be examined to determine if loading alters previous conclusions about controller efficacy.
8. Real-time evaluations should be performed on other manipulators to experimentally determine if PUMA specific results can be readily extended.
9. Modifications to the hardware level commands permit multiple non-identical manipulators to be networked by a series of RHCS links. The MIT manipulator in the RAL has been connected by a RHCS link to the hierarchy [71]. Theories for coordinated motion of two manipulators should now be evaluated and a multi-manipulator controller performance baseline established.

While those areas are being investigated research should commence on the task of integrating the hierarchical robotic evaluation environment with the vision and gripping systems to develop a platform for the evaluation of theories for hierarchically based intelligent machines.

APPENDIX A: Additional Fast IC1 Figures

TABLE A.1a APPENDIX A DATA KEY

TITLE = XCTISMT

X - Test type.

R - RAL inertial parameters

T - TARN inertial parameters

CT - Dynamic model identifer

NI - Newton-Euler with actuator inertias

FI - Full inertia with actuator inertias

DI - Diagonal inertia with actuator inertias

4F - Hybrid full inerita with actuator inertias

4D - Hybrid diagonal inerita with actuator inertias

CT - Computed-torque algorithm identifier

12 - Diagonal inertia dynamics

13 - Full inertia dynamics

14 - Block inertia dynamics

I - Initial condition specifier

0 - IC0 (0,-90,90,0,1,0,)

1 - IC1 (0,-135,135,0,1,0)

2 - IC2 (90,0,0,90,90,90)

S - Trajectory speed specifier

0 - Slow speed

1 - Fast speed

M - External load specifier

0 - unloaded

1 - fully loaded(2.3kg)

T - Sampling time specifier

2 - 14ms

3 - 21ms

TABLE A.1b APPENDIX A SYMBOL KEY

Figure A.1 Symbol Key

□	XNI1102
▶	XFI1102
○	XDI1102

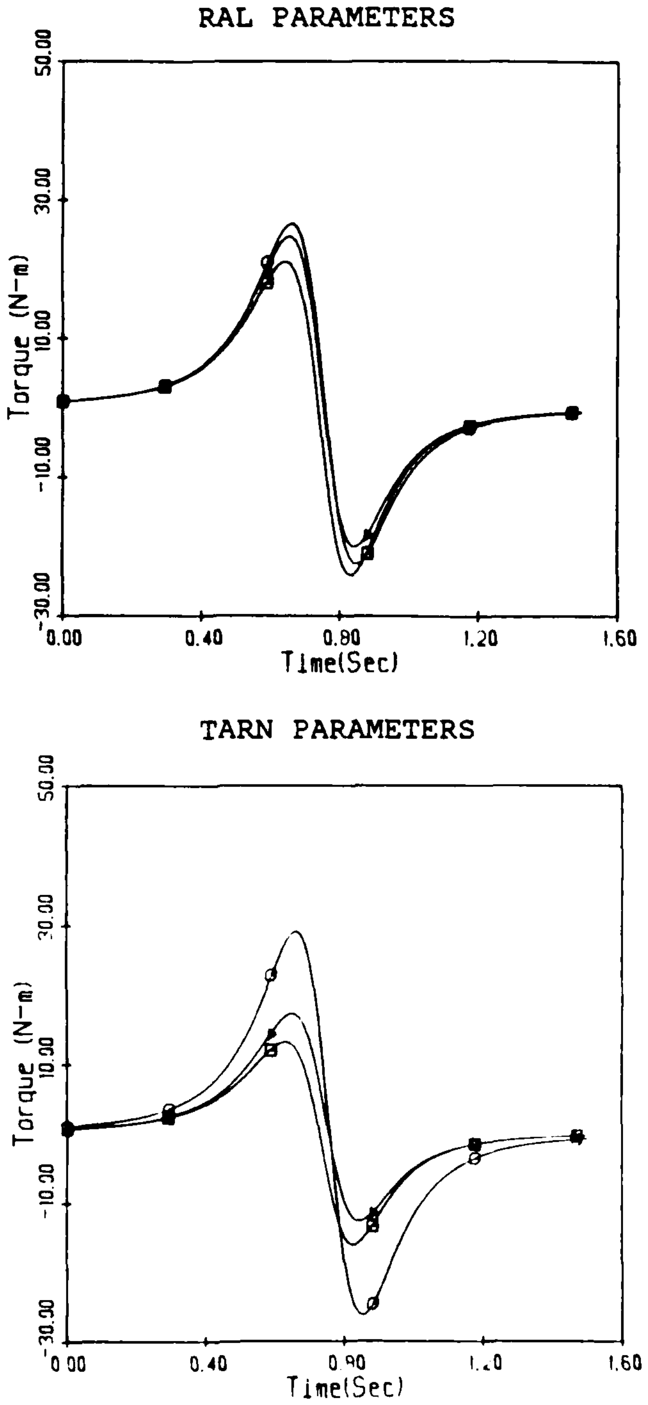
Figure A.2 Symbol Key

▶	TFI1102
○	TDI1102
◇	T4F1102
□	T4D1102

Figure A.3 and A.4 Symbol Key

○	X121102
▶	X131102
◇	X141102

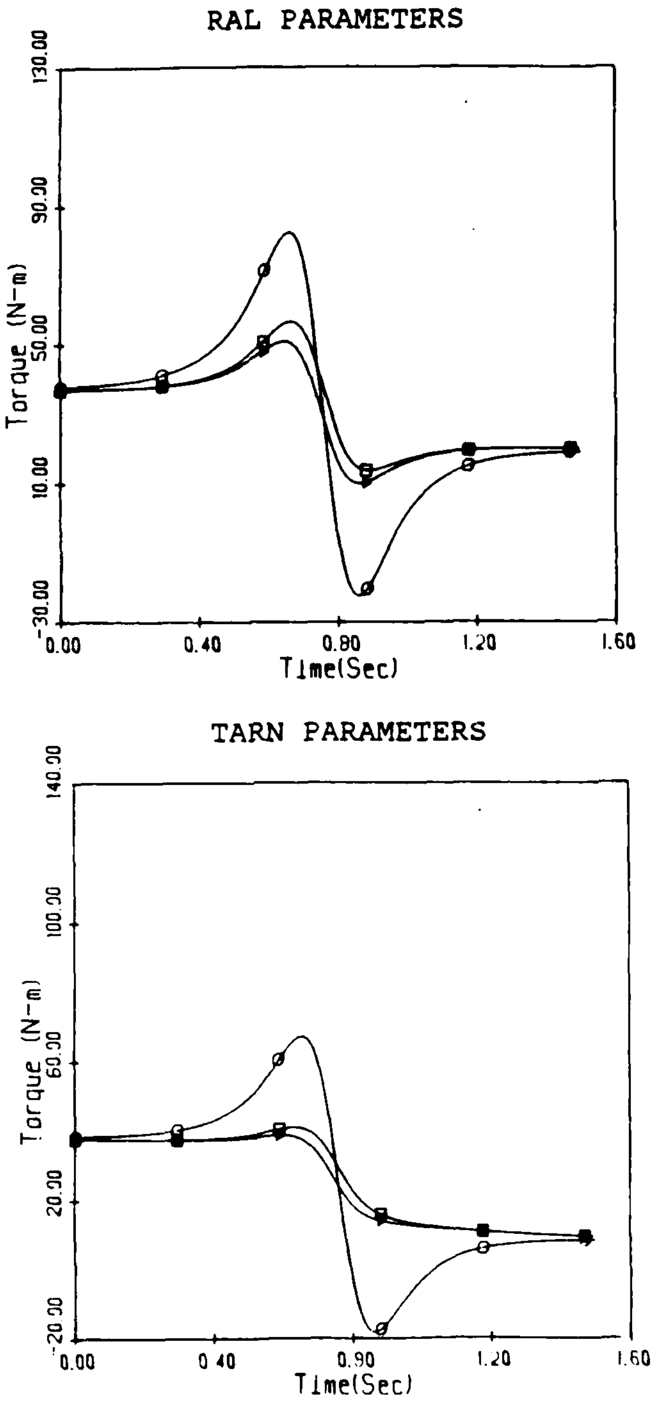
OPEN-LOOP TORQUE COMPARISON
INERTIAL PARAMETER EVALUATION


RAL PARAMETERS

TARN PARAMETERS

Figure A.1a Joint 1 Fast IC1 Open-loop Torques

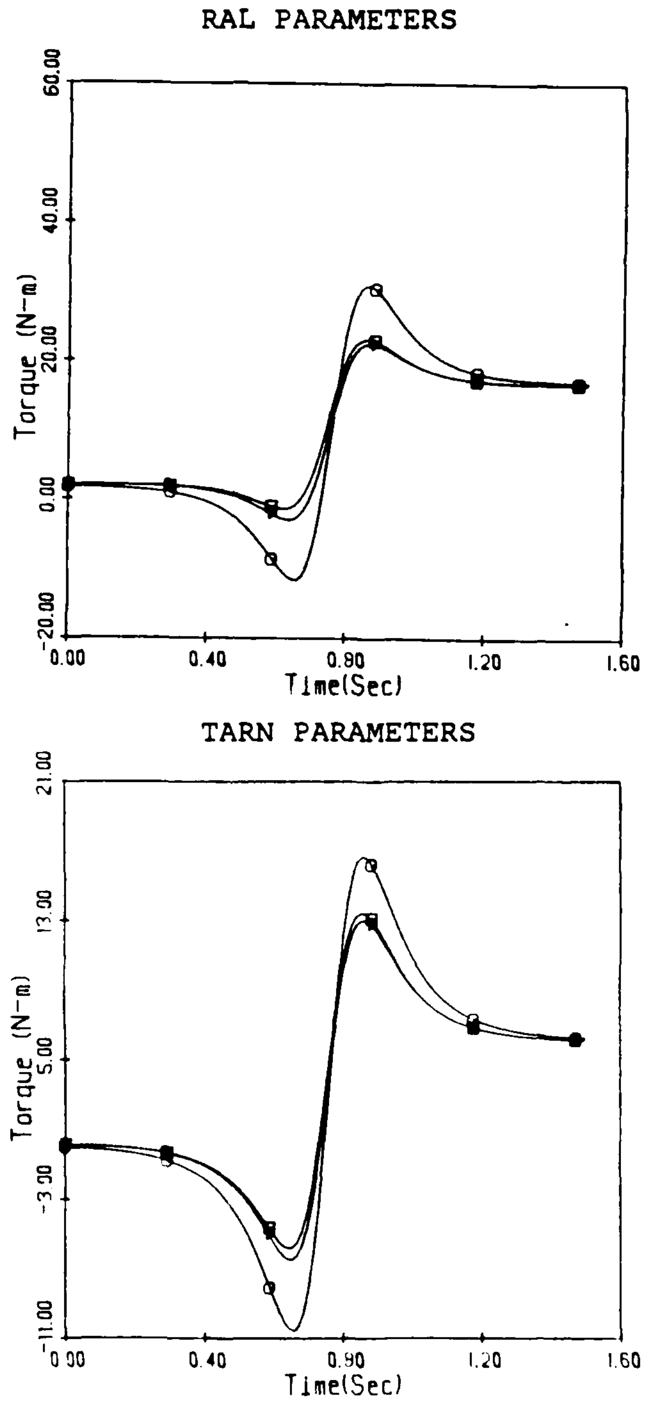
OPEN-LOOP TORQUE COMPARISON
INERTIAL PARAMETER EVALUATION


RAL PARAMETERS

TARN PARAMETERS

Figure A.1b Joint 2 Fast IC1 Open-loop Torques

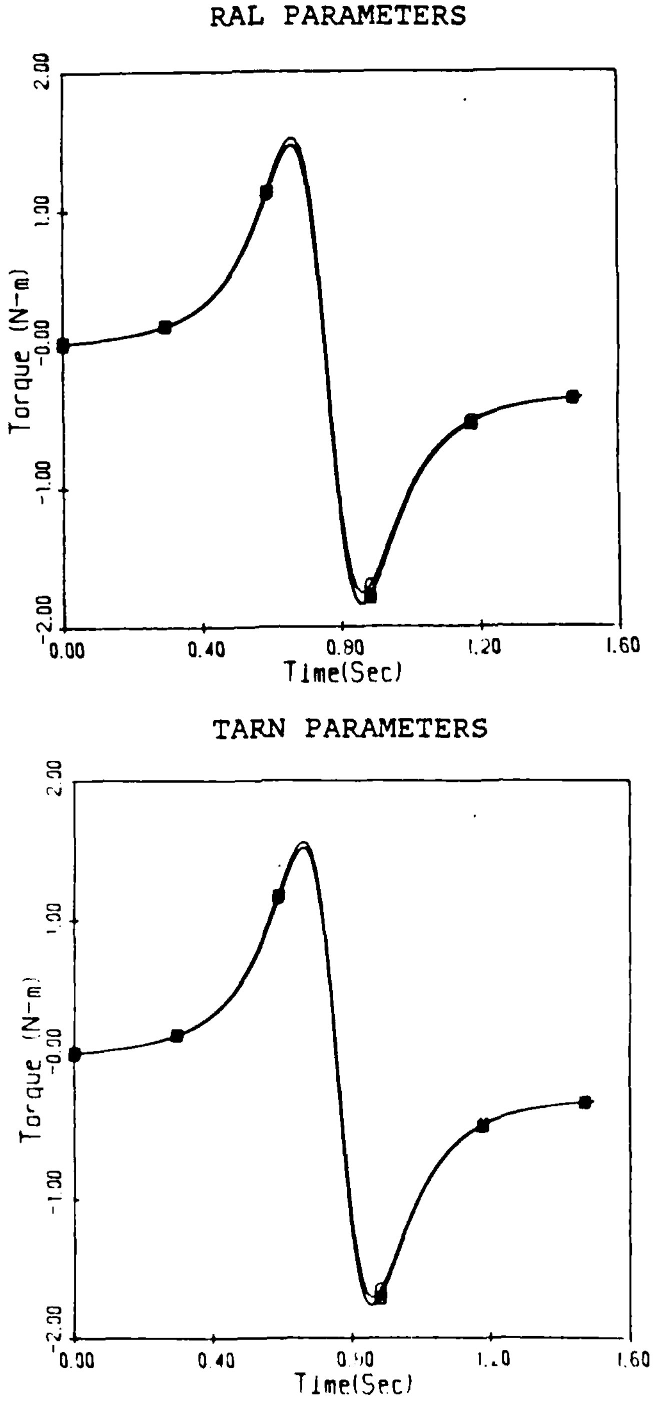
OPEN-LOOP TORQUE COMPARISON
INERTIAL PARAMETER EVALUATION


RAL PARAMETERS

TARN PARAMETERS

Figure A.1c Joint 3 Fast IC1 Open-loop Torques

OPEN-LOOP TORQUE COMPARISON
INERTIAL PARAMETER EVALUATION

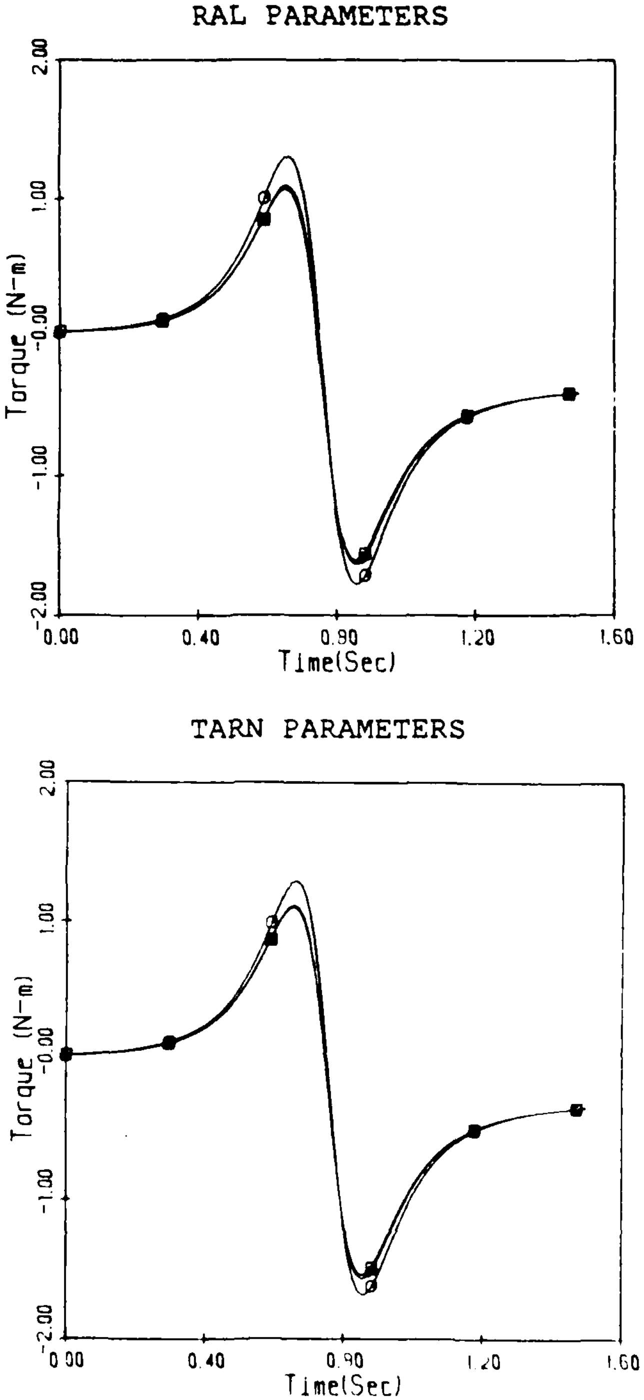


RAL PARAMETERS

TARN PARAMETERS

Figure A.1d Joint 4 Fast IC1 Open-loop Torques

OPEN-LOOP TORQUE COMPARISON
INERTIAL PARAMETER EVALUATION

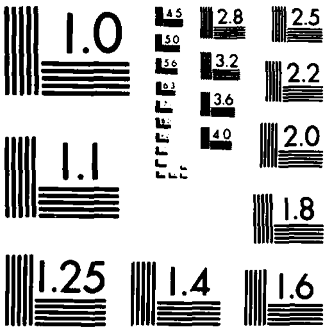

RAL PARAMETERS

TARN PARAMETERS

Figure A.1e Joint 5 Fast IC1 Open-loop Torques

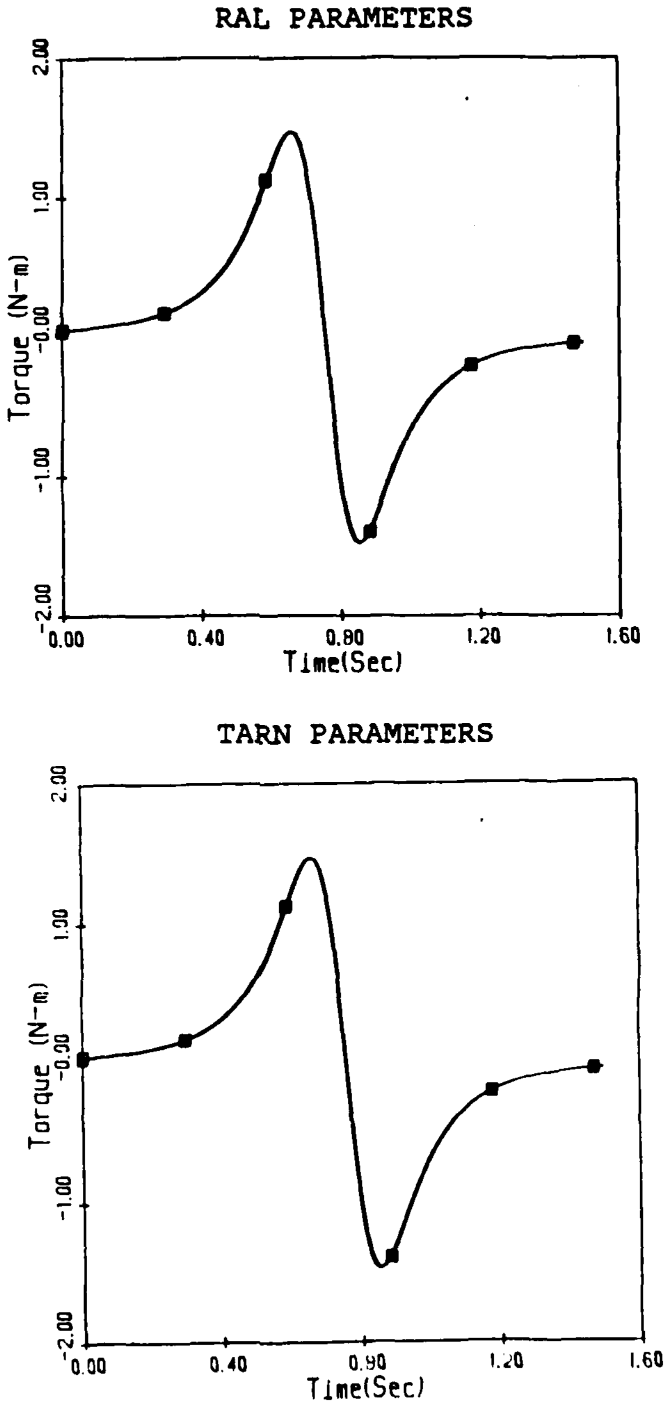
AD-A172 919 ROBOTIC MANIPULATOR CONTROL PERFORMANCE EVALUATION(U) 3/3
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH
M B LEAHY AUG 86 AFIT/CI/NR-86-173D

UNCLASSIFIED F/G 13/9 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

OPEN-LOOP TORQUE COMPARISON
INERTIAL PARAMETER EVALUATION



RAL PARAMETERS

TARN PARAMETERS

Figure A.1f Joint 6 Fast IC1 Open-loop Torques

OPEN-LOOP TORQUE COMPARISON
TARN vs HYBRID

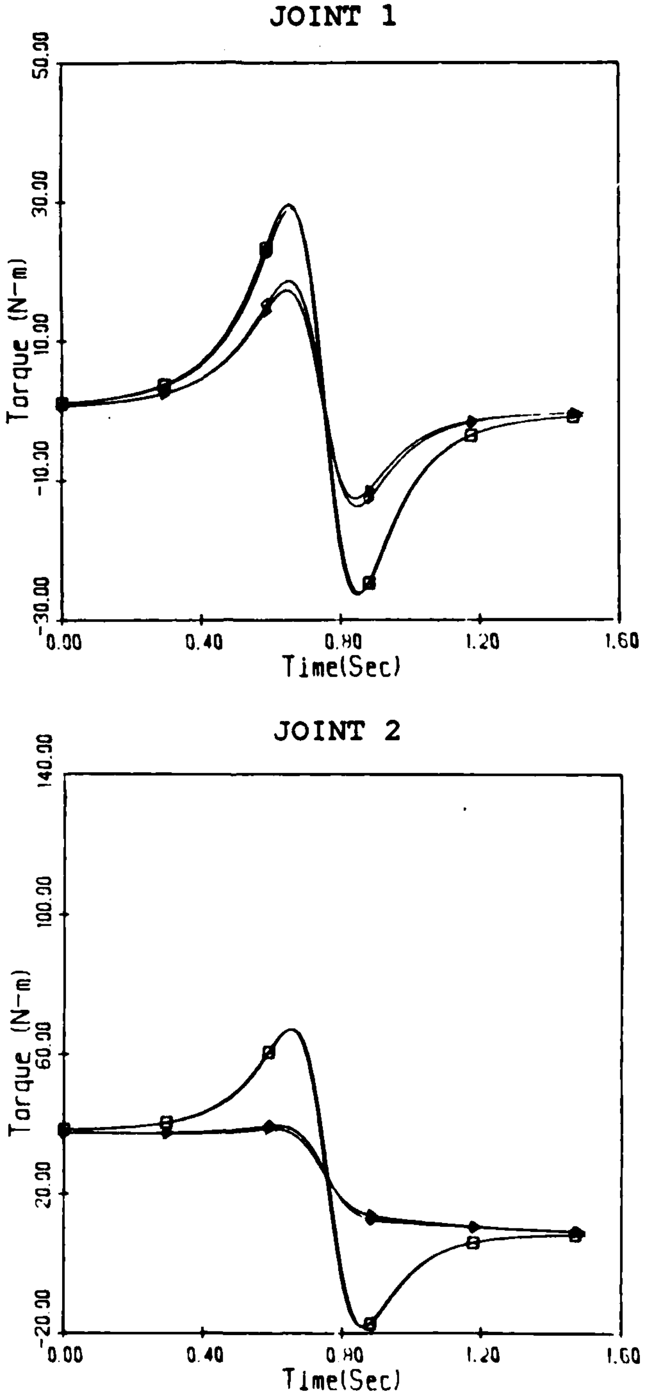
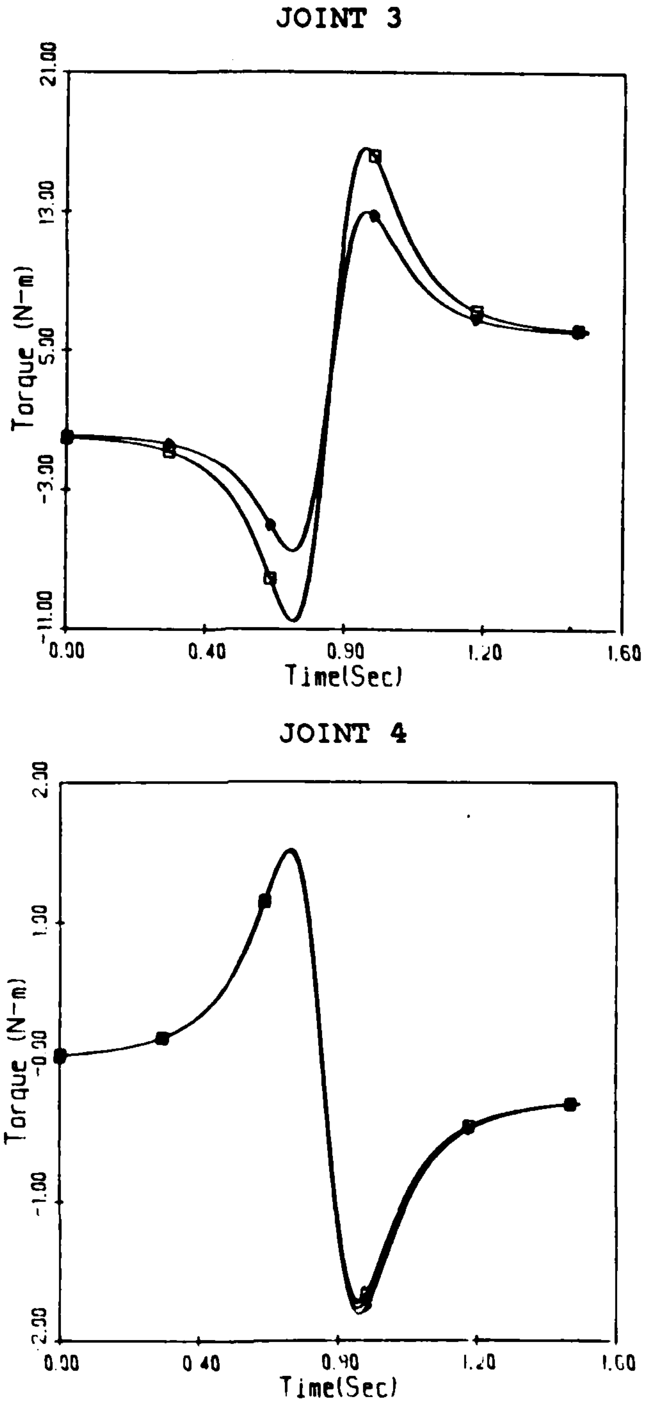


Figure A.2a Fast IC1 Open-loop Hybrid Torques

OPEN-LOOP TORQUE COMPARISON
TARN vs HYBRID

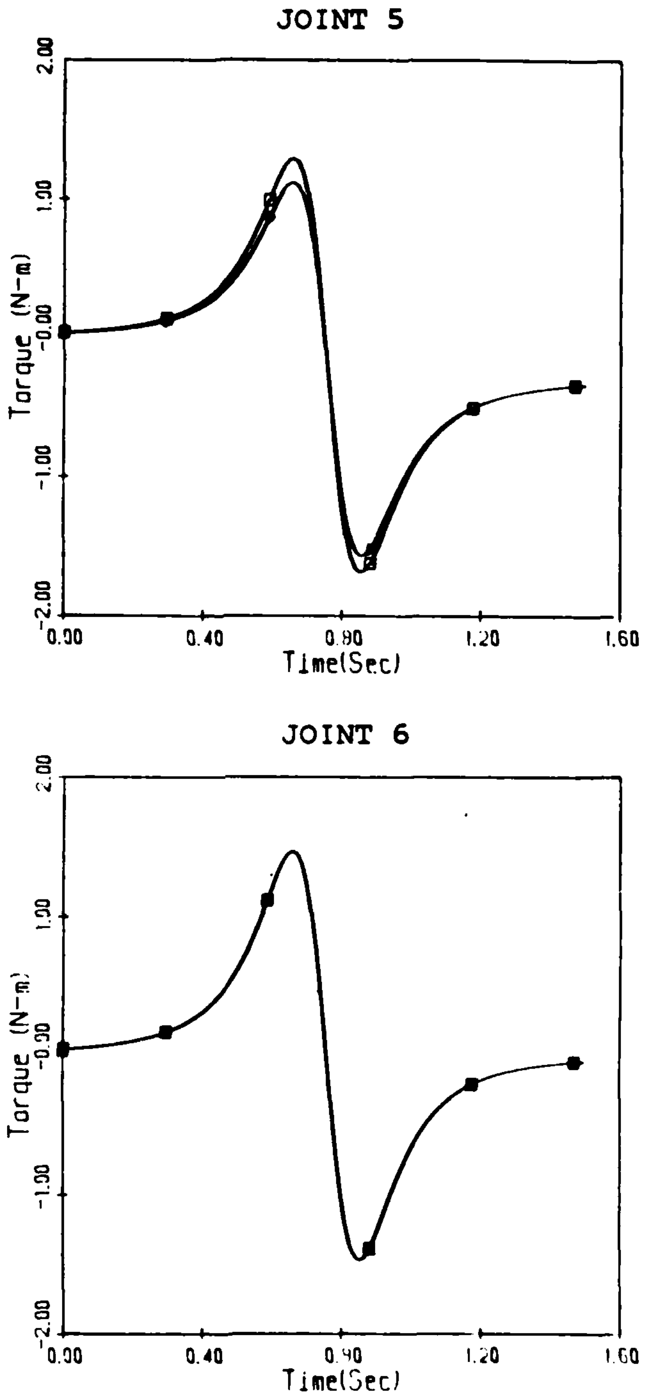


JOINT 3

JOINT 4

Figure A.2b Fast IC1 Open-loop Hybrid Torques

OPEN-LOOP TORQUE COMPARISON
TARN vs HYBRID

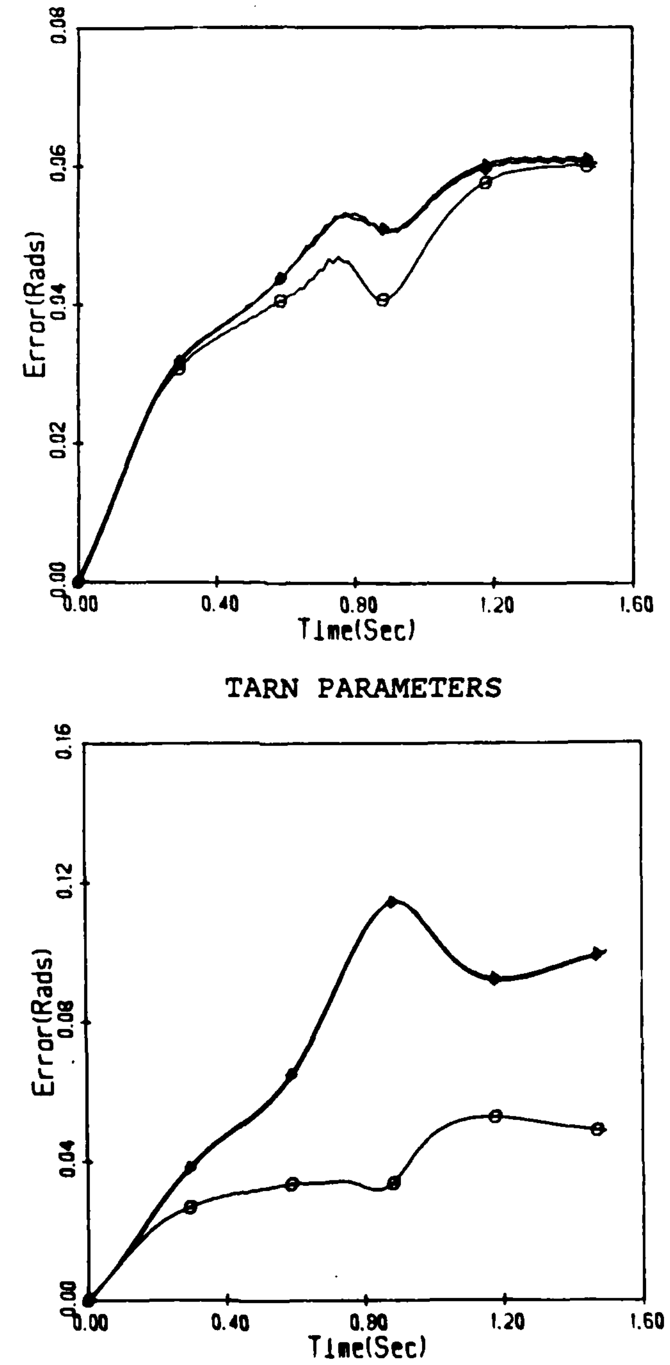


JOINT 5

JOINT 6

Figure A.2c Fast IC1 Open-loop Hybrid Torques

PERFORMANCE COMPARISON
INERTIAL PARAMETER EVALUATION

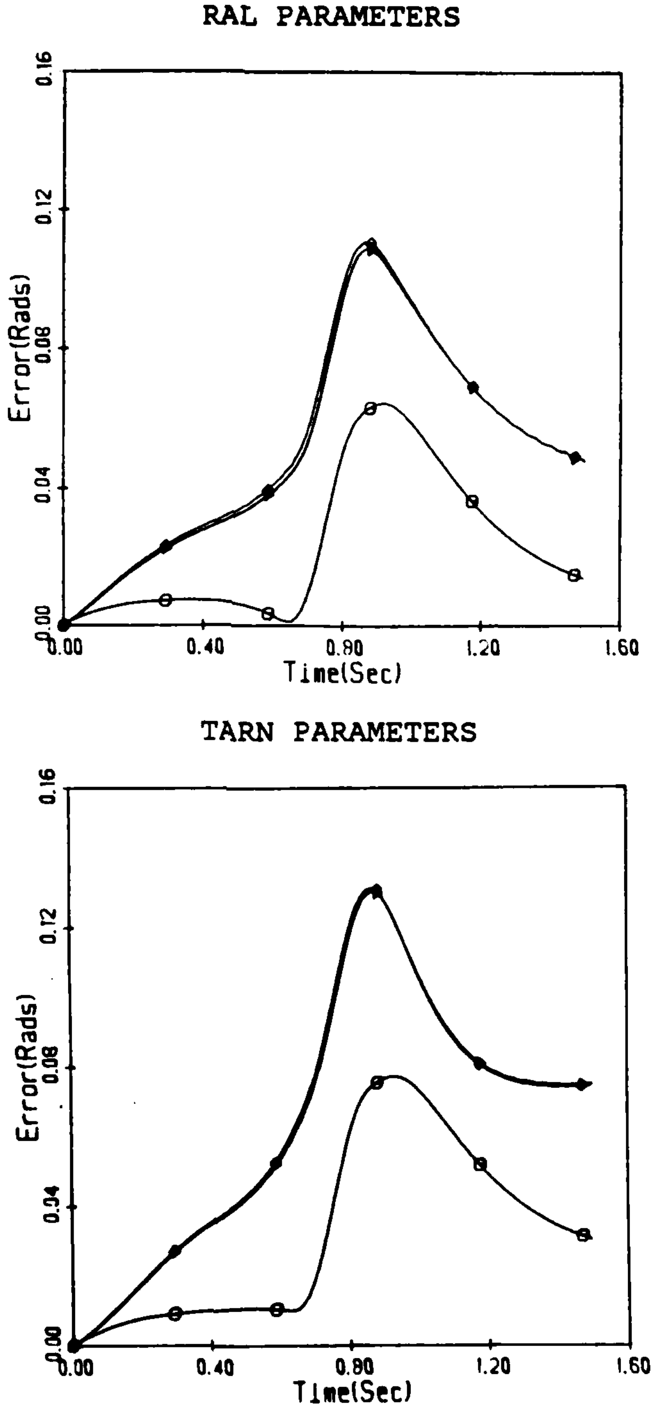


RAL PARAMETERS

TARN PARAMETERS

Figure A.3a Joint 1 Fast IC1 Position Error

PERFORMANCE COMPARISON
INERTIAL PARAMETER EVALUATION

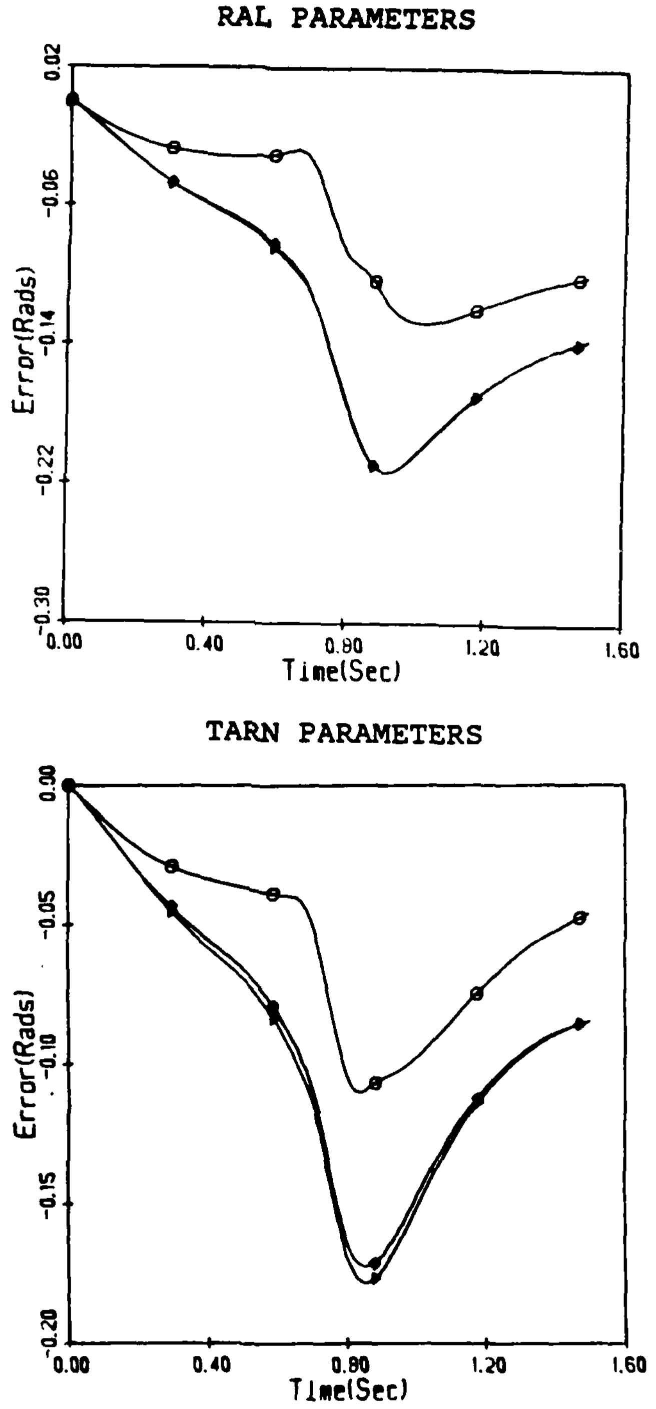


RAL PARAMETERS

TARN PARAMETERS

Figure A.3b Joint 2 Fast IC1 Position Error

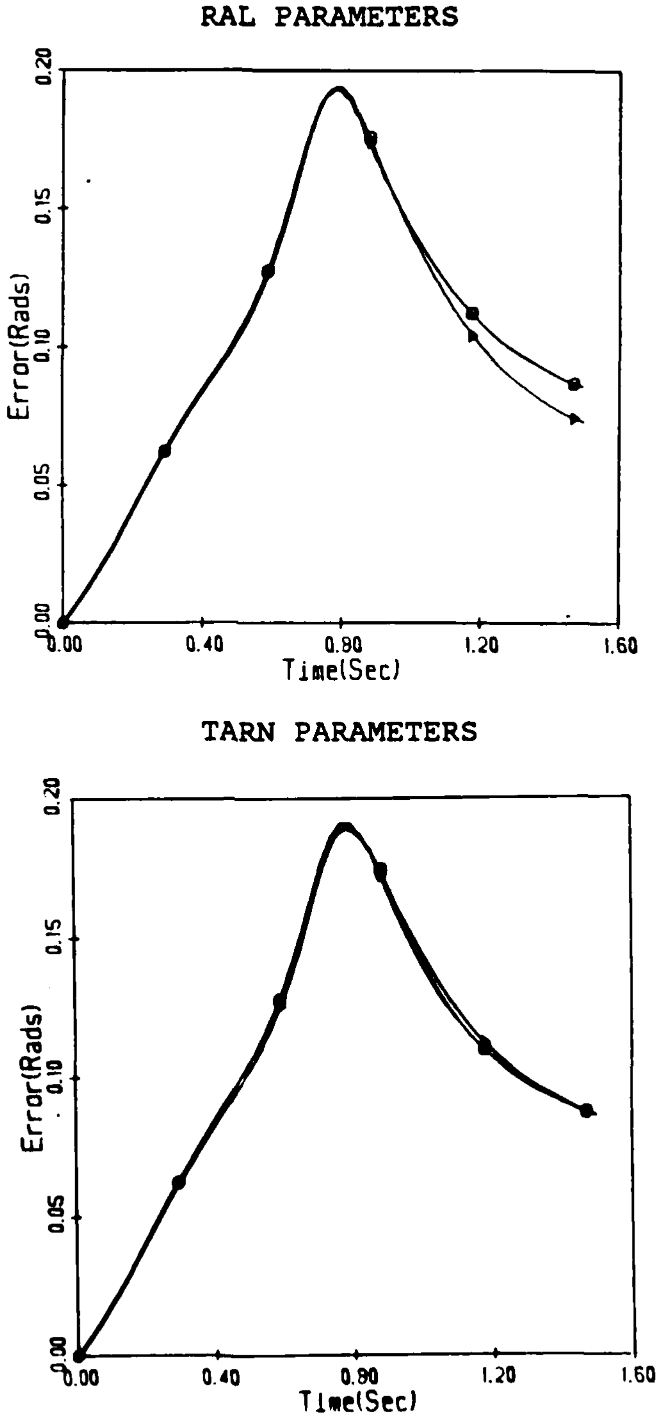
PERFORMANCE COMPARISON
INERTIAL PARAMETER EVALUATION


RAL PARAMETERS

TARN PARAMETERS

Figure A.3c Joint 3 Fast IC1 Position Error

PERFORMANCE COMPARISON
INERTIAL PARAMETER EVALUATION

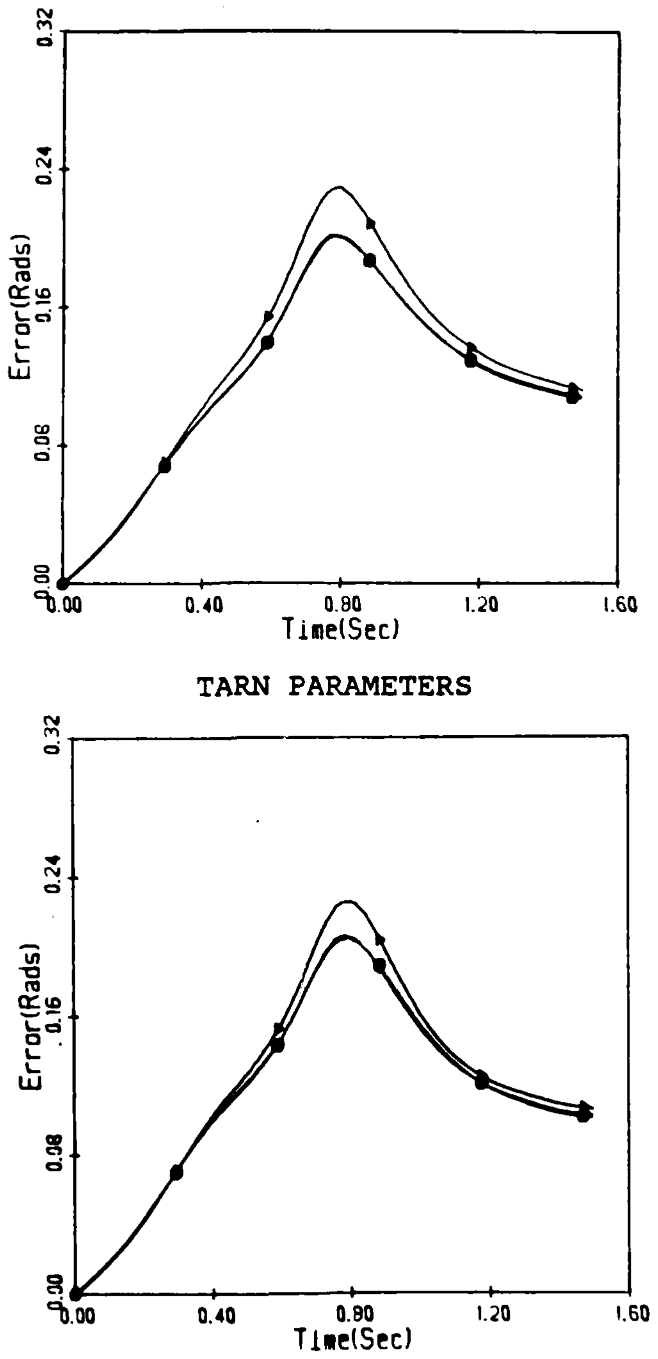


RAL PARAMETERS

TARN PARAMETERS

Figure A.3d Joint 4 Fast IC1 Position Error

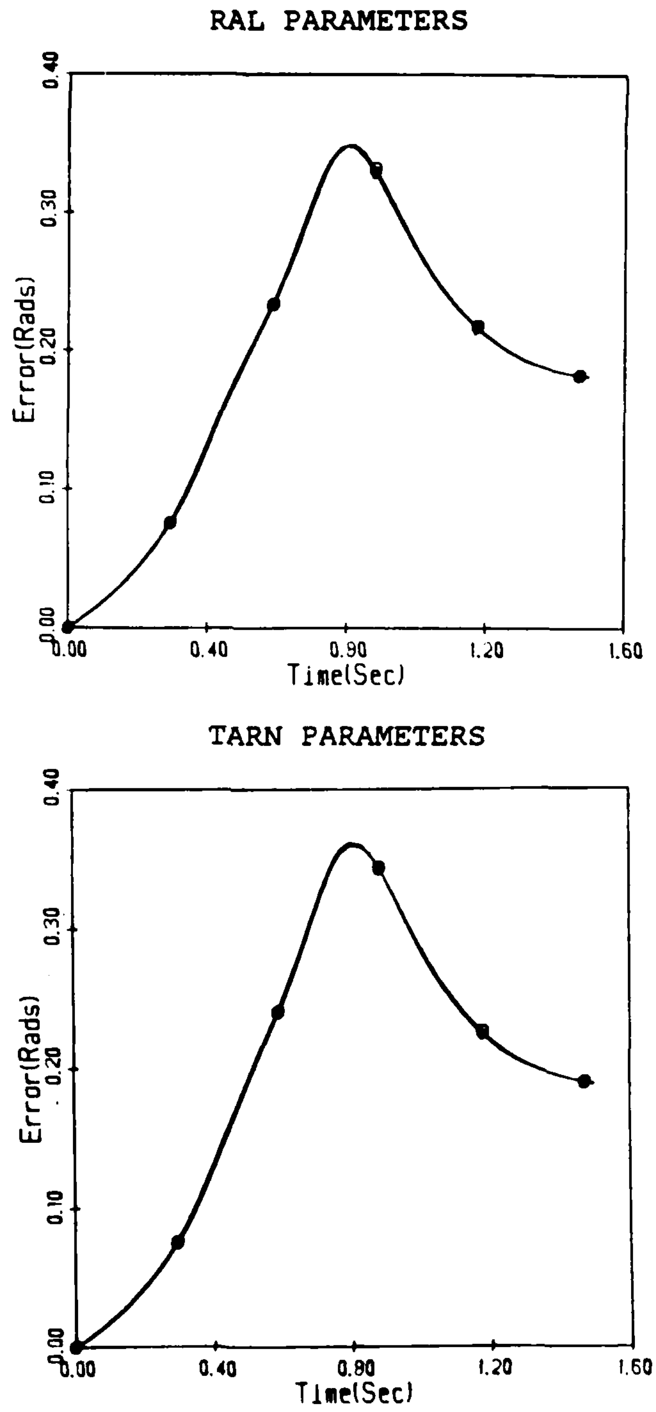
PERFORMANCE COMPARISON
INERTIAL PARAMETER EVALUATION


RAL PARAMETERS

TARN PARAMETERS

Figure A.3e Joint 5 Fast IC1 Position Error

PERFORMANCE COMPARISON
INERTIAL PARAMETER EVALUATION

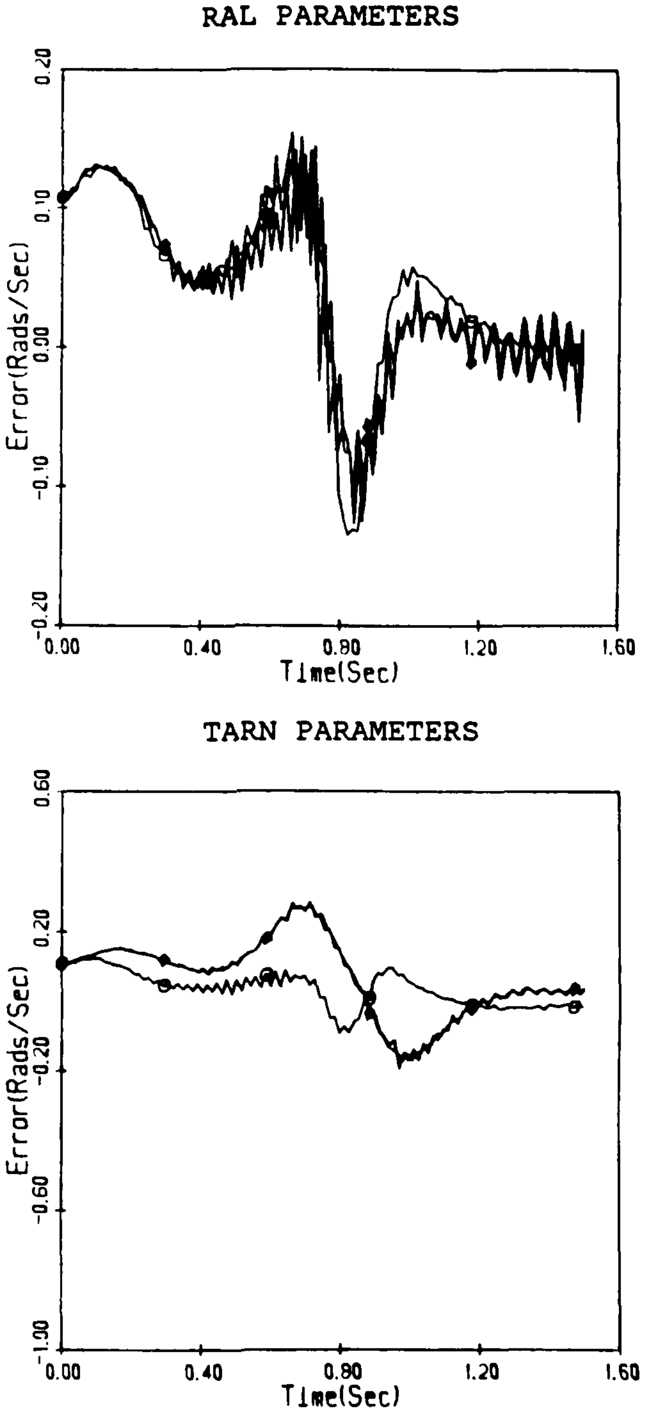


RAL PARAMETERS

TARN PARAMETERS

Figure A.3f Joint 6 Fast IC1 Position Error

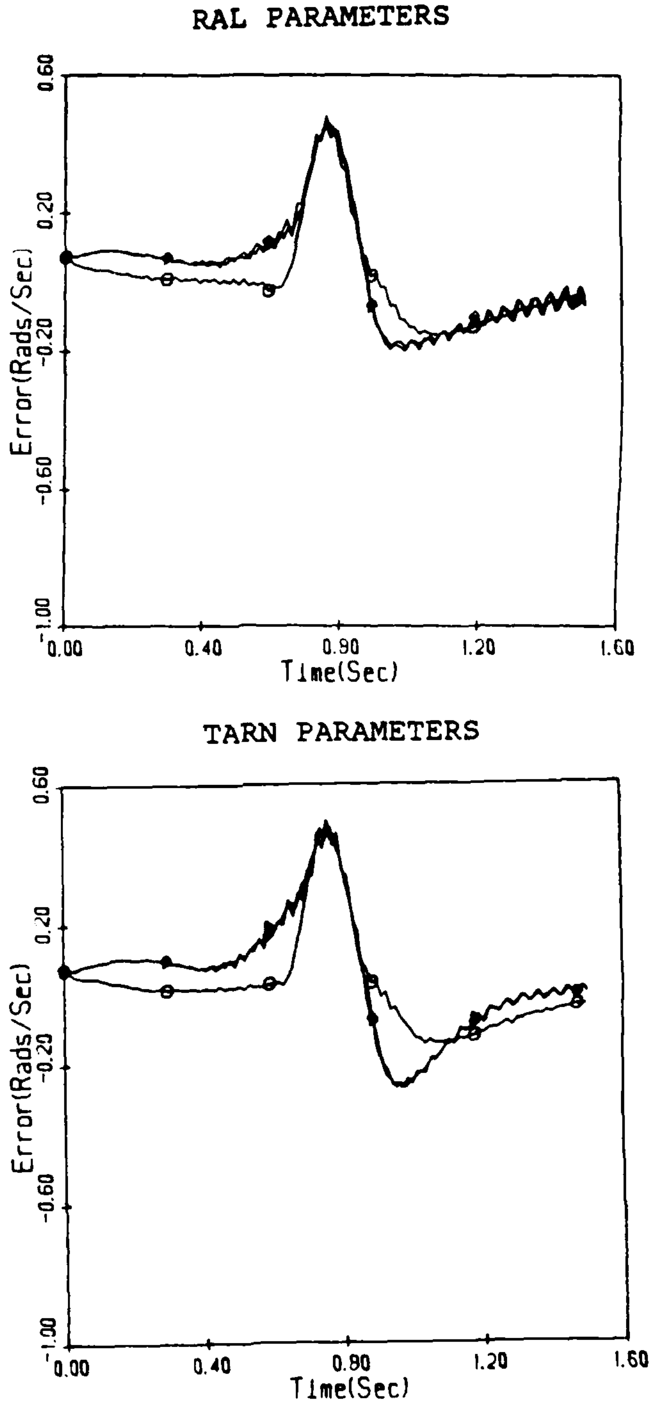
PERFORMANCE COMPARISON
INERTIAL PARAMETER EVALUATION


RAL PARAMETERS

TARN PARAMETERS

Figure A.4a Joint 1 Fast IC1 Velocity Error

PERFORMANCE COMPARISON
INERTIAL PARAMETER EVALUATION



RAL PARAMETERS

TARN PARAMETERS

Figure A.4b Joint 2 Fast IC1 Velocity Error

PERFORMANCE COMPARISON
INERTIAL PARAMETER EVALUATION

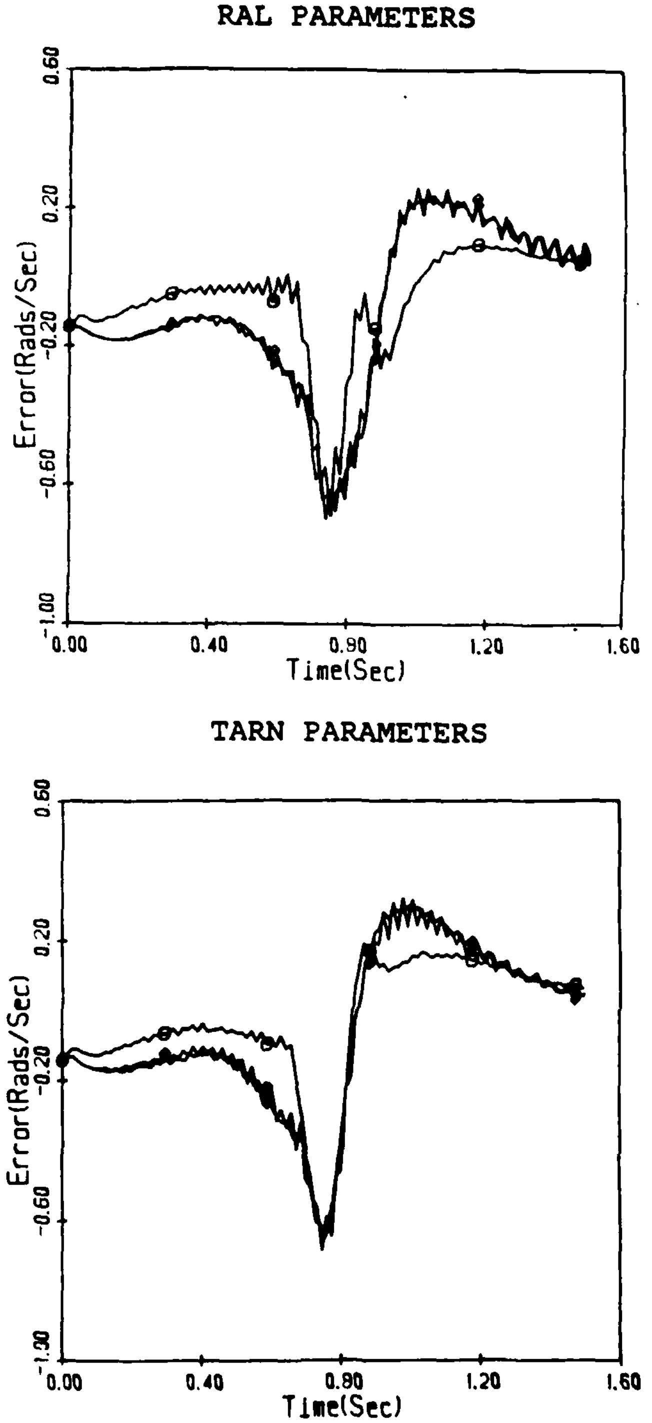
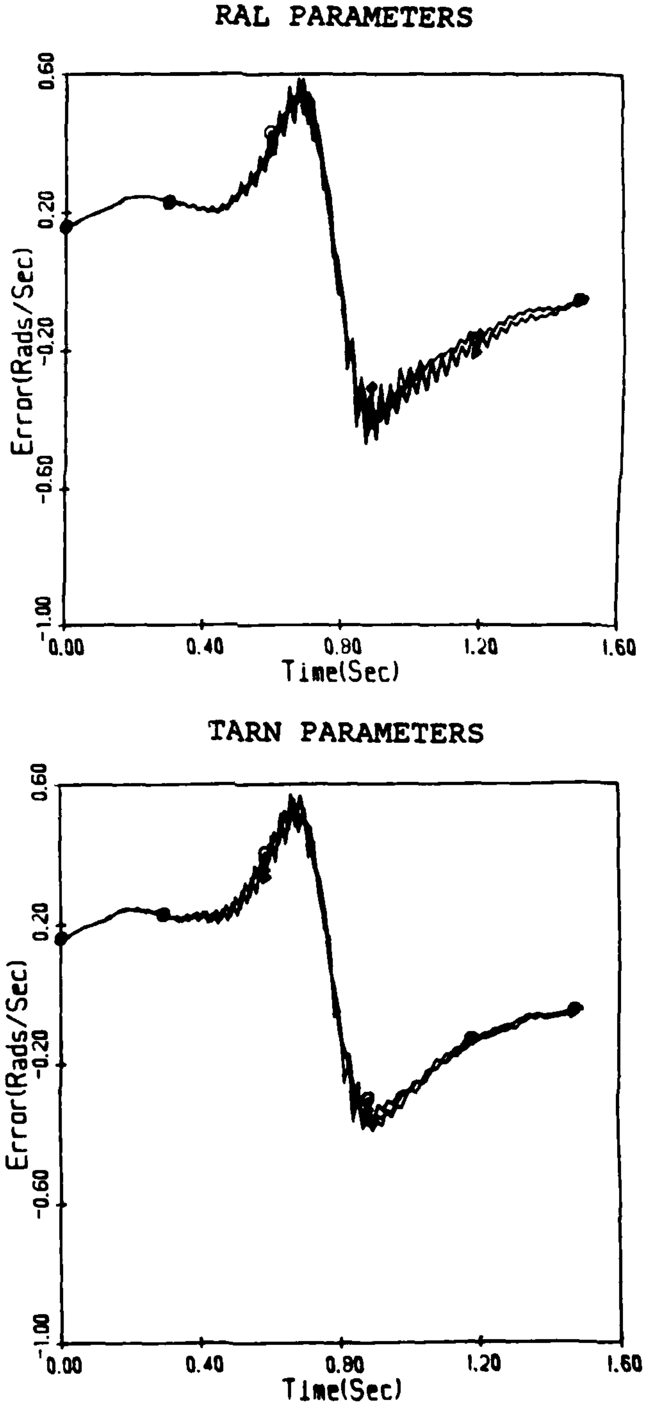


Figure A.4c Joint 3 Fast IC1 Velocity Error

PERFORMANCE COMPARISON
INERTIAL PARAMETER EVALUATION

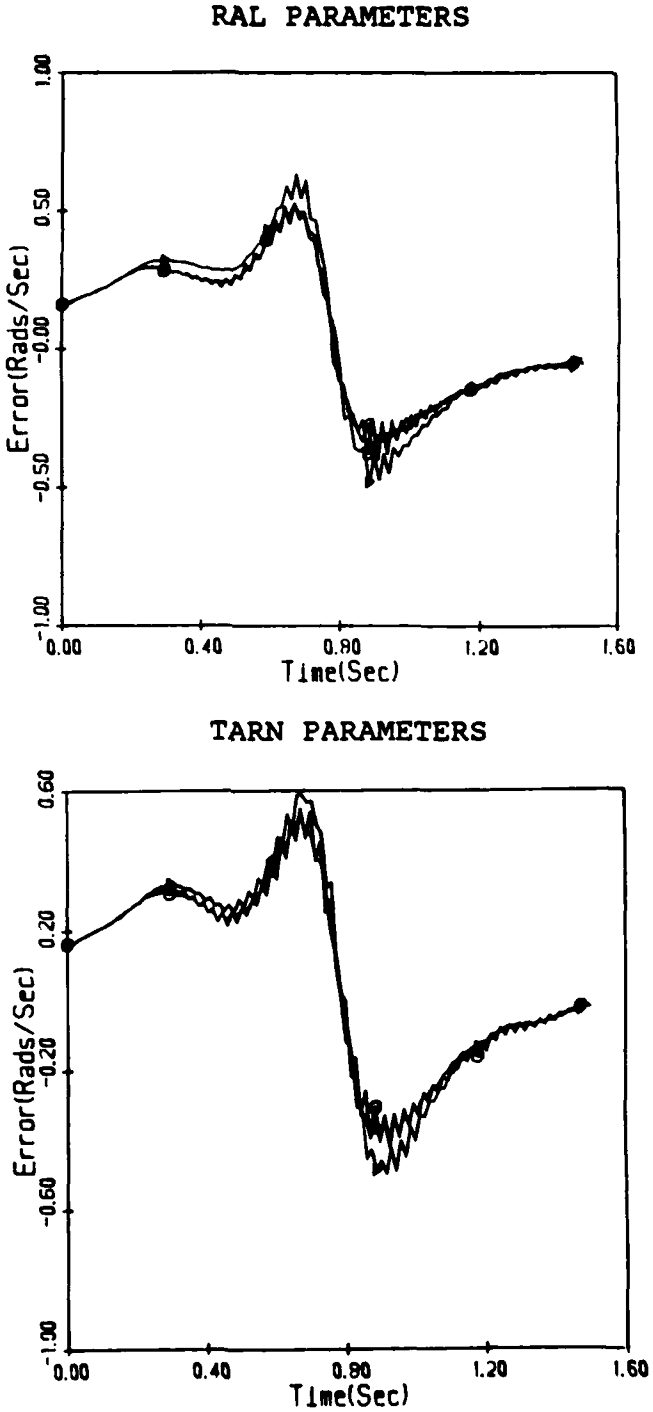


RAL PARAMETERS

TARN PARAMETERS

Figure A.4d Joint 4 Fast IC1 Velocity Error

PERFORMANCE COMPARISON
INERTIAL PARAMETER EVALUATION

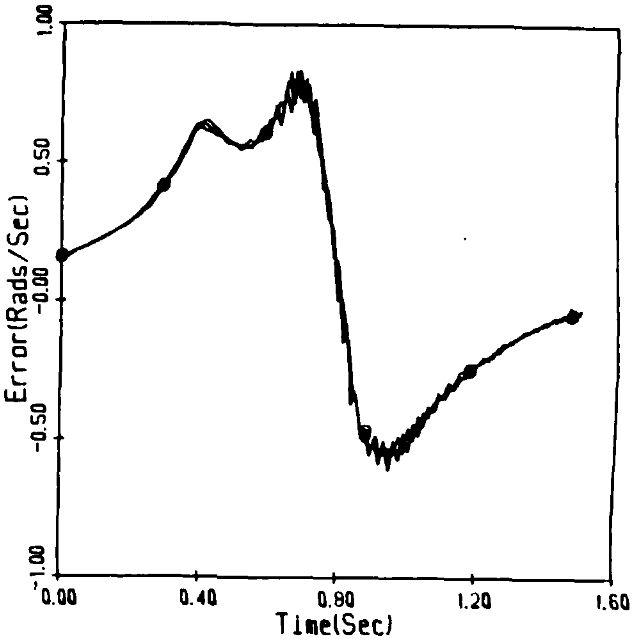

RAL PARAMETERS

TARN PARAMETERS

Figure A.4e Joint 5 Fast IC1 Velocity Error

PERFORMANCE COMPARISON
INERTIAL PARAMETER EVALUATION

RAL PARAMETERS



TARN PARAMETERS

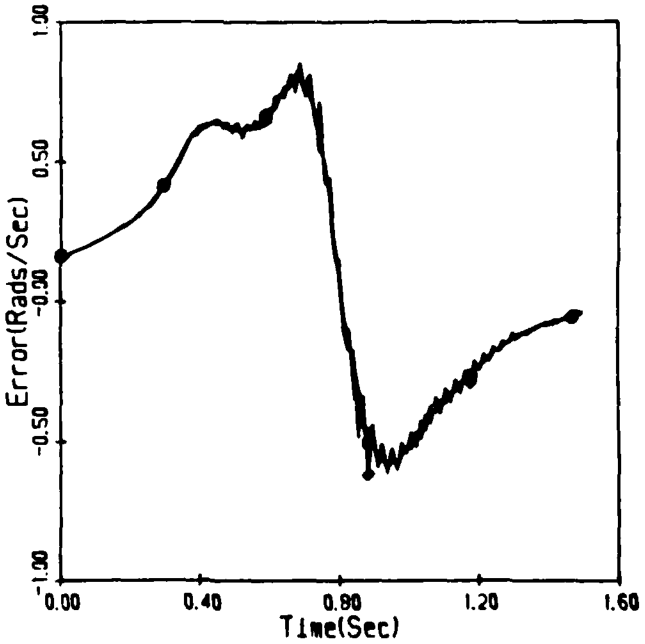


Figure A.4f Joint 6 Fast IC1 Velocity Error

LITERATURE CITED

1. Alford, C. O. and Belyeu, S. M., "A Computer Control Structure for Coordination of Two Robot Arms," Proceedings of the American Control Conference, San Diego, CA, pp. 880-1, June 6-8, 1984.

2. Alford, C. O. and Belyeu, S. M., "Coordinated Control of Two Robot Arms," Proceedings of the International Conference on Robotics, Atlanta, GA, pp. 468-73, 1984.

3. An, C. H., Atkeson, C. G. and Hollerbach, J. M., "Experimental Determination of the Effect of Feedforward Control on Trajectory Tracking Errors," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 55-60, San Francisco, CA, April 1986.

4. Anex, R. P. and Hubbard, M., "Modelling and Adaptive Control of a Mechanical Manipulator," Proceedings of the American Control Conference, San Diego, Ca., pp. 1237-42, June 6-8, 1984.

5. Backes, P. G., Leininger, G. G. and Chung, C., "Real-Time Cartesian Coordinate Hybrid Control of a PUMA 560 Manipulator," Proceedings of the IEEE International Conference on Robotics and Automation, St. Louis, MO, pp. 608-13, March 26-28, 1985.

6. Barbera, A. J., Fitzgerald, M. L., Albus, J. S. and Haynes, L. S., "RCS: The NBS Real-Time Control System," Presented at the Robots 8 Conference and Exposition, Detroit, Michigan, June 1984.

7. Bejczy, A. K. and Lee, S., "Generalized Bilateral Control of Robot Arms," Proceedings of the American Control Conference, San Diego, Ca., pp. 1883-91, June 6-8, 1984.

8. Bejczy, A. K. and Lee, S., "Robot Arm Dynamical Reduction for Control," Proceedings of the 22nd IEEE Conference on Decision and Control, San Antonio, TX, pp. 1466-76, December 16, 1983.

9. Bejczy, A. K. and Paul, R. P., "Simplified Robot Arm Dynamics for Control," Proceedings of the 20th IEEE Conference on Decision and Control, San Deigo, CA, pp. 261-2, December 16-18, 1981.

10. Bejczy, A. K., Tarn, T. J. and Chen, Y. L., "Robot Arm Dynamic Control by Computer," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 960-70, St. Louis, MO, March 25-8, 1985.

11. Brooks, R. A., "Symbolic Error Analysis and Robot Planning," The International Journal of Robotics Research, Vol. 1, No. 4, pp. 29-68, Winter 1982.

12. Cesareo, G., Nicolo, F. and Nicosia, S., "DYMIR: A Code for Generating Dynamic Models of Robots," Proceedings of the International Conference on Robotics, Atlanta, GA, pp. 115-20, March 13-15, 1984.

13. Chen, J. B., Fearing, R. S., Armstrong, B. S. and Burdick, J. W., "NYMPH: A Multiprocessor for Manipulation Applications," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1731-6, San Francisco, CA, April 1986.

14. Chung, M. J. and Lee, C. S. G., "An Adaptive Control Strategy for Computer-based Manipulators," Report RSD-TR-10-82, University of Michigan College of Engineering at Ann Arbor, August 1982.

15. Chung, C. H. and Leininger, G., "Adaptive Self-Tuning Control of Manipulator in Task Coordinate System," Proceedings of the International Conference on Robotics, Atlanta, GA, pp. 530-9, March 13-15, 1984.

16. Dubowsky, S., "On the Adaptive Control of Robotic Manipulators: The Discrete-Time Case," Proceedings of the Joint Automatic Control Conference, Charlottesville, Va., June 17-19, 1981.

17. Dubowsky, S. and Desforges, D. T., "The Application of Model-Referenced Adaptive Control to Robotic Manipulators," Journal of Dynamic Systems, Measurement and Control, Vol. 101, pp. 193-200, September 1979.

18. Dupourque, V., "A Robot Operating System," Proceedings of the International Conference on Robotics, Atlanta, GA, pp. 342-8, March 13-15, 1984.

19. Foulloy, L. P. and Kelley, R. B., "Improving the Precision of a Robot," Proceedings of the International Conference on Robotics, Atlanta, GA, pp. 62-7, March 13-15, 1984.

20. Freund, E., "Fast Nonlinear Control with Arbitrary Pole Placement for Industrial Robots and Manipulators," Intl. Journal of Robotics Research, Vol. 1 (1982) pp. 65-78.

21. Furuta, K., Kosuge, K. and Yamakita, M., "Trajectory Tracking Control of Robot Arms Using ORBIX," Journal of Robotic Systems, Vol. 2, No. 1, pp. 89-112, 1985.

22. Fussell, P. S., Wright, P. K. and Bourne, D., "A Design of a Controller as a Component of a Robotic Manufacturing System," Journal of Manufacturing Systems, Vol. 3, No. 1, pp. 1-11, 1984.

23. Gallager, R. G., Information Theory and Reliable Communication, J. Wiley, 1968.

24. Goor, R. M., "A New Approach to Robot Control," Proceedings of the American Control Conference, pp. 385-9, Boston MA., 1985.

25. Gruver, W. A., Soroka, B. I., Craig, J. J. and Turner, T. L., "Industrial Robot Programming Languages: A Comparative Evaluation," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-14, No. 4, pp. 565-70, July/August 1984.

26. Hayward, V. and Paul, R. P. C., "Introduction to RCCL: A Robot Control "C" Library," Proceedings of International Conference on Robotics, Atlanta, GA, pp. 293-7, March 13-15, 1984.

27. Hearn, A. C., Reduce 2 User's Manual, March 1973.

28. Hollerbach, J.M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Cooperative Study of Dynamics Formulation Complexity," IEEE Transactions of Systems, Man, Cybernetics, vol SMC-10, No.11, pp. 730-736, November, 1980.

29. Hollerbach, J. M. and Sahar, G., "Wrist-Partitioned Inverse Kinematic Accelerations and Manipulator Dynamics," Proceedings of the International Conference on Robotics, Atlanta, GA, pp. 152-61, March 13-15, 1984.

30. Horak, D. T., "A Fast Computational Scheme for Dynamic Control of Manipulators," Proceedings of the American Control Conference, San Diego, Ca., pp. 1883-91, June 6-8, 1984.

31. Horowitz, R. and Tomizuka, M., "An Adaptive Control Scheme for Mechanical Manipulators - Compensation of Nonlinearity and Decoupling Control," ASME Paper No. 80-WA/DSC-6.

32. Isaguiree, A., Paul, R. P. C.,"Computation of the

Inertial and Gravitational Coefficients of the Dynamics Equations for a Robot Manipulator with a Load," Proceedings of the International Conference on Robotics, St. Louis, MO, pp. 1024-31, March 26-28, 1985.

33. Kanade, T., Khosla, P. K. and Tanaka, N., "Real-Time Control of CMU Direct-Drive Arm II Using Customized Inverse Dynamics," Proceedings of the 23rd IEEE Conference on Decision and Control, Las Vegas, NV, pp. 1345-52, December 1984.

34. Khosla, P. K. and Kanada, T., "Experimental Evaluation of the Feedforward Compensation and Computed-Torque Control Schemes," Proceedings of the American Control Conference, pp. 790-8, Seatle, WA, June 1986.

35. Khosla, P. K. and Kanade, T., "Real-Time Implementation and Evaluation of Model-Based Controls on CMU DD ARM II," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1546-55, San Francisco, CA, April 1986.

36. Leahy, M. B. Jr., "A Hierarchical Computer Control System for the PUMA-600 Robot Arm," RAL Technical Report No. 49, RPI, January, 1985.

37. Leahy, M. B. Jr., "RAL Hierarchical Robotic Evaluation Environment Software Manual Volume 1: VAX FORTRAN Algorithms," RAL Technical Report No. 72, May 1986.

38. Leahy, M. B. Jr., "RAL Hierarchical Robotic Evaluation Environment Software Manual Volume 2: PUMA Dynamics Algorithms," RAL Technical Report No. 74, May 1986.

39. Leahy, M. B. Jr., "RAL Hierarchical Robotic Evaluation Environment Software Manual Volume 3: Assembly Language Algorithms," RAL Technical Report No. 74, May 1986.

40. Leahy, M. B. Jr., "R3AGE: The RAL Real-Time Robotic Algorithm Exerciser User's Guide Version 1.0," RAL Technical Report No. 61, November 1985.

41. Leahy M. B. Jr., "The RAL Hierarchical Control System," RAL Technical Report No. 67, RPI, March 1986.

42. Leahy, M. B. Jr., "The RAL Robotic Simulation Environment User's Guide," RAL Technical Report in work.

43. Leahy, M. B. Jr., "Performance Characterization of a PUMA-600 Robot," RAL Technical Report No. 56, RPI, September, 1985.

44. Leahy, M. B. Jr., Nugent, L. M., Valavanis, K. P. and Saridis, G. N., "Efficient Dynamics for A PUMA-600," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 519-24, San Francisco, CA, April 1986.

45. Leahy, M. B. Jr., Nugent, L. M., Valavanis, K. P. and Saridis, G. N., "Efficient Puma Manipulator Jacobian Calculation and Inversion," RAL Research Report No. 76, RPI, June 1986, submitted to the Journal of Robotic Systems.

46. Leahy, M. B. Jr. and Saridis, G. N., "Compensation of Unmodeled Manipulator Forces," Technical Report No. 77, RPI, June 1986.

47. Leahy, M. B. Jr. and Saridis, G. N., "The Effects of Unmodeled Forces on Robot Control," to be published at the 25th CDC, Athens, Greece, Dec. 1986.

48. Leahy, M. B. Jr., and Saridis, G. N., "The RAL Hierarchical Control System," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 407-11, San Francisco, CA, April 1986.

49. Leahy, M. B. Jr., Saridis, G. N. and Valavanis, K. P., "Evaluation of Dynamic Models for Robot Control," RAL Technical Report No. 57, RPI, May 1986, submitted to the IEEE Journal of Robotics and Automation.

50. Leahy M. B. Jr. and Valavanis, K.P., "A VAX Computer-Controlled PUMA-600 Robot Arm Software Manual," RAL Technical Report No. 33, April 1984.

51. Leahy M. B. Jr. and Valavanis, K. P., "Software for Real-Time control of a VAX-11/750 Computer Controlled PUMA-600 Robot Arm," RAL Technical Report No. 37, RPI, July, 1984.

52. Leahy, M. B. Jr., Valavanis, K. P. and Saridis, G. N., "The Effects of Dynamics Models on Robot Control," Pro- of the IEEE International Conference on Robotics and Automation, pp. 49-54, San Francisco, CA, April 1986.

53. Lee, C. S. G. and Chen, M. H., "A Suboptimal Control Design for Mechnical Manipulators," Proceedings of the 1983 American Control Conference, San Francisco, CA, pp. 1056-61, June 22-24, 1983.

54. Lee, C. S. G., Chung, M. J. and Lee, B. H., "Adaptive Control for Robot Manipulator in Joint and Cartesian

Coordinates," Proceedings of International Conference on Robotics, Atlanta, GA, pp. 530-9, March 13-15, 1984.

55. Lee, C. S. G., Chung, M. J. and Turney, J. L., "On the Control of Mechanical Manipulators," Technical Report RSD-TR-5-82, Center for Robotics and Integrated Manufacturing, University of Michigan at Ann Arbor.

56. Lee, G. S. G., Gonzalez, R. C. and Fu, K. S., Tutorial on Robotics, IEEE Computer Society Press, 1983.

57. Lee, C. S. G., Mudge, T. N. and Turney, J. L., "Hierarchical Control Structure Using Special Purpose Processors for the Control of Robot Arms," Proceedings of the 1982 Conference on Pattern Recongnition and Image Processing, pp. 634-40, 1982.

58. Leininger, G. G., Backes, P. G. and Chung, C., "Tool Coordinate Control of a PUMA Robot Arm," Proceedings of the American Control Conference, San Diego, Ca., pp. 1560-5, June 6-8, 1984.

59. Lem, E., "Development of a Robotic System Software," University of Toronto, Bachelor Thesis, April 1984.

60. Lozinski, C. A., "Robot Calibration: Method and Results," Master's Thesis, MIT, June 1984.

61. Luh, J. Y. S., "An Anatomy of Industrial Robots and Their Controls," IEEE Transactions on Automatic Control, Vol. AC-28, No. 2, pp. 133-53, February 1983.

62. Luh, J. Y. S. and Lin, C. S., "Scheduling of Parallel Computation for a Computer-Controlled Mechanical Manipulator," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-12, No. 2, March/April 1982.

63. Luh, J. Y., Walker, M. W., and Paul, R. P. C., "On-Line Computational Scheme for Mechnical Manipulators," Transactions of ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 120, pp. 69-76, June 1980.

64. Luh, J. Y. S., Walker, M. W. and Paul, R. P. C., "Resolved-Acceleration Control of Mechanical Manipulators," IEEE Transactions on Automatic Control, Vol. AC-25, No. 3, pp. 468-474, June 1980.

65. Luo, G. L., and Saridis G. N.,"L-Q Design of PID Controllers for Robot Arms," IEEE Journal of Robotics and Automation, Vol. RA-1, No. 3, September 1985.

66. Luo, G. L. and Saridis, G. N., "Optimal/PID Formulation

of Robotic Manipulator Control," RAL Technical Report No. 15, RPI, March 1983.

67. McCain, H. G., "A Hierarchically Controlled, Sensory Interactive Robot in the Automated Manfacturing Research Facility," Proceedings of the International Conference on Robotics, St. Louis, MO, pp. 931-9, March 26-28, 1985.

68. Microcomputers and Memories, Digital Equipment Corporation, 1982.

69. Moanga, A., Manputianu, D., Oprea, E., Nedelea, D, and Udrea, B., "Advanced Control and Programming System for Industrial Robots," 14th ISIR Gothenberg, Sweden, pp. 487-96, October 2-4, 1984.

70. Murkerjee, A. and Ballard, D. H., "Self-Calibration in Robot Manipulators," Proceedings of the International Conference on Robotics, St. Louis, MO, pp. 1050-7, March 26-28, 1985.

71. Murphy, S, "MIT RHCS User's Guide Version 1.0," RAL Technical Report No. 66, RPI, August 1986.

72. Murray, J. J. and Neuman, C. P., "ARM: An Algebraic Robot Dynamic Modeling Program," Proceedings of the International Conference on Robotics, Atlanta, GA, pp. 103-14, March 13-15, 1984.

73. Nigam, R. and Lee, C. S. G., "A Multiprocessor-Based Controller for the Control of Mechanical Manipulators," Proceedings of the International Conference on Robotics, St. Louis, MO, pp. 815-21, March 26-28, 1985.

74. Nugent L. M., "Software Used for Reduction of PUMA-600 Jacobian, Inverse Jacobian and Lagrange-Euler Dynamics Calculations," RAL Technical Report No. 64, RPI, January 1986.

75. Orin, D. E., Chao, H. H., Olson, K. W. and Schrader, W. W., "Pipeline/Parallel Algorithms for the Jacobian and Inverse Dynamics Computations," Proceedings of the International Conference on Robotics, St. Louis, MO, pp. 785-9, March 26-28, 1985.

76. Ozguner, F. and Kao, M. L., "A Reconfigurable Multiprocessor Architecture for Reliable Control of Robotic Systems," Proceedings of the nternational Conference on Robotics, St. Louis, MO, pp. 802-6, March 26-28, 1985.

77. Paul, R. P., Robot Manipulators, MIT Press, 1981.

78. Paul, R. P. C., "Modeling, Trajectory Calculation, and Servoing of a Computer Controlled Arm," Stanford AI Lab Memo Am-177, November 1972.

79. Paul, R. P., Rong, M. and Zhang, H.,"The Dynamics of A PUMA Manipulator," Proceedings of The American Control Conference, San Francisco, CA, pp. 491-496, June 22-24, 1983.

80. Pfeffer, L., Khatib, O. and Hake, J., "Joint Torque Sensory Feedback in the Control of a PUMA Manipulator," Proceedings of the American Control Conference, pp. 818-24, Seattle, WA., June 18-20, 1986.

81. PDP-11 Microcomputer Interfaces Handbook, Digital Equipment Corporation, 1983.

82. Salisbury, J. K., "Active Stiffness Control of a Manipulator in Cartesian Coordinates," Proceedings of the 19th IEEE Conference on Decision and Control, Albuq., NM, pp. 95-100, Dec. 1980.

83. Sanderson, A. C., "Parts Entropy Methods for Robotic Assembly System Design," Proceedings of the International Conference on Robotics, Atlanta, GA, pp. 600-8, March 13-5, 1984.

84. Saridis, G. N., Advances in Automation and Robotics Volume 1, JAI Press Inc, 1985.

85. Saridis, G. N., "Intelligent Control-Operating Systems in Uncertain Environments," Uncertainty And Control, Vol. 70, J. Ackermann, Editor, Springer-Verlag, May 1985.

86. Saridis, G. N., "Intelligent Robotic Control," IEEE Transactions on Automatic Control, Vol. AC-28, No. 5, pp. 547-57, May 1983.

87. Saridis, G. N. and Blumberg, J., "Entropy as a Cost Criterion for Hierarchical Intelligent Control," RAL Technical Report No. 17, August 1983.

88. Saridis, G. N. and Lee, C. S. G., "An Approximation Theory of Optimal control for Trainable Manipulators," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-9, No. 3, March 1979, pp. 152-9.

89. Schwan, K., Bihari, T., Weide, B. and Taulbee, G., "GEM: Operating System Primitives for Robots and Real-Time Control Systems," Proceedings of the International Con-

ference on Robotics, St. Louis, MO, pp. 807-13, March 26-28, 1985.

90. Shimano, B., "VAL: A Versatile Robot Programming and Command System," COMPSAC 79 Conference Proceedings, pp. 878-83.

91. Shimano, B. E., Geschke, C. C. and Spalding, C. H. III, "VAL-II: A New Robot Control System for Automatic Manufacturing," Proceedings of International Conference on Robotics, Atlanta, GA, pp. 278-91, March 13-15, 1984.

92. Stone, H. W., Sanderson, A. C. and Neuman, C. P., "Arm Signature Identification," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 41-8, San Francisco, CA, April 1986.

93. Sweet, L. M., and Good , M. C., "Redefinition of the Robot Motion-Control Problem," IEEE Control Systems Magazine, Vol. 5. No. 3, 1985.

94. Tarn, T. J., Bejczy, A. K., Hau S., and Yun X., "Inertia Parameters of PUMA 560 Robot Arm," Robotics Laboratory Report SSM-RL-85-01, Washington University, St. Louis, Mo., September 1985.

95. Tarn, T. J., Bejczy, A. K., Isidori, A., and Chen, Y., "Nonlinear Feedback in Robot Arm Control," Pr›ceedings of the 23rd CDC, pp. 736-51, Las Vegas, NV, Dec 1984.

96. Tarn, T. J., Bejczy, A. K. and Yun X., "Dynamic Equations for PUMA 560 Robot Arm," Robotics Laboratory Report SSM-RL-85-02, Washington University, St. Louis, Mo., July 1985.

97. Taylor, R. H., Summers, P. D. and Meyer, J.M., "AML: A Manufacturing Language," The Internal Journal of Robotics Research, Vol. 1, No. 3, pp. 19-41, Fall 1982.

98. Tourassis, V. D. and Neuman, C. P., "Robust Feedback of an Articulated Robot: A Case-Study," Proceedings of the 24th CDC, Ft. Lauderdale, Fl., pp. 1505-9, December 1985.

99. Turner, T. L., Craig, J. J. and Gruver, W. A., "A Microprocessor Architecture for Advanced Robot Control," 14th ISIR, Gothenberg, Sweden, pp. 407-16, October 2-4, 1984.

100. Unimate PUMA Robot Volume 1 - Technical Manual, Unimation Inc., April 1980.

101. Unimation PUMA Robot Manual Volume II, Unimation Inc.,

1980.

102. Valavanis, K. P., "Controlling the PUMA-600 With And/Or Without Using VAL," RAL Technical Report 16, May 1983.

103. Valavanis, K. P., "On The Real-Time Control of A VAX-11/750 Computer-Controlled PUMA-600 Robot Arm," Masters Thesis, RPI, June 1984.

104. Valavanis, K. P., Leahy, M. B. Jr. and Saridis, G. N., "Real-time Evaluation of Robotic Control Methods," Proceedings of the International Conference on Robotics, St. Louis, MO, pp.644-9, March 26-28, 1985.

105. Valavanis, K. P. and Saridis, G. N., "Mathematical Formulation of the Organizer Level of an Intelligent Machine," submitted for publication.

106. VAX-11 RSX Compatibility Mode Reference Manual, Digital Equipment Corporation, September 1984.

107. Whitney, D. E., "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators," Transactions of ASME, Journal of Dynamic Systems, Measurement and Control," pp. 303-9, December 1982.

108. Whitney, D. E., Lozinski, C. A. and Rourke, J. M., "Industrial Robot Forward Calibration Method and Results," Transactions of ASME, Journal of Dynamic Systems, Measurement and Control," pp. 1-8, March 1986.

109. Wu, C-H. and Paul, R. P. C., "Resolved Motion Force Control of Robot Manipulators," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-12, No. 3, pp. 266-75, June 1982.

110. Youcef-Toumi, K. and Asada, H., "The Design of Open-Loop Manipulator Arms with Decoupled and Configuration-Invariant Inertia Tensors," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2018-26, San Francisco, CA.,April 1986.

111. Zhang, H. and Paul, R. P. C., "Hybrid Control of Robot Manipulators," Proceedings of the International Conference on Robotics, St. Louis, MO, pp. 602-7, March 26-28, 1985.

112. Zheng, Y. F. and Chen, B. R., "A Multiprocessor for Dynamics Control of Multilink Systems," Proceedings of the International Conference on Robotics, St. Louis, MO, pp. 295-300, March 26-28, 1985.

END

11-86

DTIC